COMPUTATIONAL METHODS FOR THE IDENTIFICATION AND QUANTIFICATION OF TRANSCRIPT ISOFORMS FROM NEXT GENERATION SEQUENCING DATA

Inauguraldissertation
zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von
FOIVOS GYPAS
aus
Griechenland

Basel, 2019

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von

Prof. Dr. Mihaela Zavolan, Dr. Petr Svoboda

Basel, den 27.02.2018

Prof. Dr. Martin Spiess
Dekan

Keep Ithaka always in your mind
— Ithaka, by C. P. Cavafy

Deticated to
my partner Maria
my sister Ioli
my mother Amalia
my father George

ABSTRACT

Most mammalian genes have multiple isoforms which are generated through the use of alternative transcription initiation sites, termination sites and internal exons. High-throughput sequencing technologies enabled the discovery and quantification of many novel RNA species including protein-coding RNAs, microRNAs, long non-coding RNAs and others. Currently, what is sequenced is mostly short reads, not full-length transcripts. Thus, computational methods are needed to reconstruct transcripts and infer their expression levels from the RNA-seq data, which is challenging, due to the many biases that are introduced during sample preparation. The main aim of my thesis was to improve approaches to isoform reconstruction and quantification. I have started by evaluating the performance of isoform quantification methods using two complementary test data sets. The first was generated by simulating short read sampling from in silico transcriptomes with known transcript abundances and second by preparing and sequencing in parallel both RNA-seq and 3'end sequencing reads from the same population of cells. Many of the benchmarked methods performed comparably well, while a few were outstanding. However, all methods produced more accurate results of gene-level estimates than commonly used count-based methods. I have set up a complementary web service that developers of isoform quantification methods can use to compare the accuracy of their approach with those that we have already surveyed. Transcript quantification methods generally start from annotated transcripts, whose abundance is then estimated. However many isoforms are still to be identified. Currently available RNA-seq-based transcript reconstruction methods are insufficiently accurate, especially in the identification of transcript 5′ or 3′ ends. A catalog of poly(A) sites in the human and mouse genomes that our group constructed contains thousands of poly(A) sites located in regions that are currently annotated as intergenic and intronic. They indicate that many transcripts are yet to be annotated. Towards this goal, we developed the Terminal Exon Characterization (TEC) tool, which uses annotated intronic poly(A) sites together with RNA-seq data to reconstruct terminal exons and associated transcript isoforms. Applying TECtool to various datasets, we identified many novel tissue-specific transcripts, particularly from testis and bone marrow. Single cell data indicate that the relatively low expression of these transcripts is not due to their being expressed at low levels in individual cells, but rather to their being expressed in smaller subpopulations of cells. Ribosome profiling data suggest that novel transcript isoforms lead to the production of new proteins. TECtool can enrich the existing transcript annotation and support an improved transcript isoform abundance estimation. These in turn are relevant for the identification of binding sites for various regulators (miRNAs, RBPs), and for the annotation of protein domains. Besides

developing novel tools, I have put much effort into their automation, in line with current efforts towards reproducibility of data analysis.

ACKNOWLEDGEMENTS

At this point, I would like to thank a few people that directly or indirectly contributed to this work. First of all, I am grateful to my supervisor Mihaela Zavolan for the opportunity she gave me to work at her group, her constant encouragement to learn new things and support to apply new ideas. She was always available and responsive, she advised me wisely and brought me back to the right track when needed. I thank her also for her support that allowed me to travel to many scientific conferences and events around the world. The last four years were a great adventure for me that helped my scientific and personal development.

I would like to thank the members of my Ph.D. committee Petr Svoboda and Gunter Meister for the time and effort they spent with my thesis and for the advice and suggestions they gave me. I am also grateful to the funding agencies that supported my research and let me collaborate with great scientists in Switzerland and abroad.

I would like to thank my colleagues for useful discussions, suggestions and constructive criticism. I would especially like to thank Alexander Kanitz who shared many of his knowledge and experience and together with me is the first co-author of the first paper presented in this thesis. I also thank Andreas J. Gruber who along with me is the first co-author of the second article presented in this thesis and he as well shared many of his knowledge with me. I also thank all co-authors of these publications. I am also grateful to the members of scicore for their help and support with the infrastructure that we use every day.

I am very happy for all the new friends I made, and I am always grateful to my good friends back in Greece and abroad.

I am forever grateful to my sister Ioli, my mother Amalia and my father George who always inspire me, trust me and are there when I need them. Without their constant support and dedication to my life, I would not be able to conduct this Ph.D.

Last but not least, I would like to thank my partner Maria, who supported me from the first moment on my decision to move to Switzerland and for her constant trust, patience, and support. She inspires me every day with her unique way of looking at things, and I am one of the luckiest persons to be able to both live and work with her.

To everyone, I have mentioned and those I have forgotten: Thank you for everything.

CONTENTS

1 INTRODUCTION 1
- 1.1 Computational analysis of high-throughput sequencing (HTS) data 2
  - 1.1.1 Preprocessing 2
  - 1.1.2 Alignment methods 3
  - 1.1.3 Quantification methods 5
  - 1.1.4 Transcript identification methods 6
  - 1.1.5 Analysis of alternative splicing isoforms 7
  - 1.1.6 Alternative polyadenylation analysis 8
  - 1.1.7 Long non-coding RNAs 8
  - 1.1.8 Visualization 9

2 COMPARATIVE ASSESSMENT OF METHODS FOR THE COMPUTATIONAL INFERENCE OF TRANSCRIPT ISOFORM ABUNDANCE FROM RNA-SEQ DATA 11
- 2.1 Abstract 11
  - 2.1.0.1 Background 11
  - 2.1.0.2 Results 11
  - 2.1.0.3 Conclusions 11
- 2.2 Background 11
- 2.3 Results 13
  - 2.3.1 Runtime and memory requirements differ substantially between tools 13
  - 2.3.2 Most methods infer transcript abundances with good accuracy even from sparse datasets 16
  - 2.3.3 Explicit modeling of transcript isoforms leads to more accurate estimation of gene expression levels than count-based methods 17
  - 2.3.4 High expression levels are more accurately estimated than low expression levels 20
  - 2.3.5 The alignment program and bias correction options have little impact on the accuracy of abundance estimates 21
  - 2.3.6 Gene/transcript structural features affect the estimates of individual methods 22
  - 2.3.7 Isoform- and gene-level estimates are consistent across biological replicates 25
  - 2.3.8 3′ end sequencing provides independent estimates of isoform abundance 26
- 2.4 Discussion 31
- 2.5 Conclusions 34
- 2.6 Methods 35
  - 2.6.1 Genomes, gene annotations, and transcriptome sequences 35
  - 2.6.2 Generation of synthetic sequencing data 37
  - 2.6.3 Preparation of sequencing libraries 37

2.6.4 Pre-processing of human and mouse RNA-seq data 38
2.6.5 Alignment of synthetic and experimentally obtained reads to genomes and transcriptomes 38
2.6.6 Analysis of 3′ end sequencing data 39
2.6.7 Estimation of transcript isoform abundance 40
2.6.8 Normalization and stratification of expression 'ground truths' and estimates 45
2.6.9 Count-based gene-level estimates of expression 46
2.6.10 Evaluating the accuracy of gene/isoform abundance estimates 47
2.6.11 Availability of supporting data 47
2.7 Acknowledgements 47
3 TERMINAL EXON CHARACTERIZATION WITH TECTOOL REVEALS AN ABUNDANCE OF CELL-SPECIFIC ISOFORMS 49
3.1 Abstract 49
3.2 Introduction 49
3.3 Results 50
3.3.1 Prevalent RNA processing at intronic poly(A) sites 50
3.3.2 TECtool identifies terminal exons from RNA-sequencing data 50
3.3.3 TECtool reproducibly and accurately identifies transcripts 53
3.3.4 TECtool identifies cell-type-specific isoforms 54
3.3.5 Previously unknown isoforms are expressed in subsets of single cells 54
3.4 Discussion 56
3.5 Methods 56
3.5.1 Datasets 56
3.5.2 Poly(A) sites 58
3.5.3 Analysis of intronic poly(A) sites identified by 3′-end processing 58
3.5.4 TECtool 58
3.5.4.1 Inputs, outputs and user options 58
3.5.4.2 Selection of intronic PASs 59
3.5.4.3 Identification of candidate novel terminal exon 59
3.5.4.4 Collection of training exonic regions 59
3.5.4.5 Feature computation 60
3.5.5 Classifier training and prediction of novel terminal exons 61
3.5.6 Novel transcripts and CDS annotation 62
3.5.7 Automated analysis of RNA-seq datasets with TECtool 62
3.5.8 Analysis of mouse RNA-sequencing data 62
3.5.9 Analysis of novel transcript expression in 32 human tissues 63
3.5.10 Visualization of read densities 63
3.5.11 Statistics 63

3.5.11.1 Analysis of single end RNA-seq data 63
3.5.11.2 Analysis of paired-end RNA-seq data 65
3.5.11.3 Analysis of single cell sequencing data 65
3.5.12 Analysis of TECtool running time 66
3.5.13 Analysis of ribosome profiling data 66
3.5.14 Transcript and terminal exon reconstruction with StringTie 67
3.5.14.1 Transcript and terminal exon reconstruction with Cufflinks 67
3.5.15 Parallel analysis of long and short read data 67
3.6 Acknowledgements 70
4 DISCUSSION 71
A BENCHMARKING SUPPLEMENTS 75
B TECTOOL SUPPLEMENTS 89
C PUBLICATIONS AND CONTRIBUTION 103
BIBLIOGRAPHY 105

LIST OF FIGURES

Figure 1.1 Analysis of HTS data 4
Figure 2.1 Running time and memory requirements 16
Figure 2.2 Influence of sequencing depth and expression levels on the accuracy of expression estimates 18
Figure 2.3 Impact of gene structural features on expression estimates 24
Figure 2.4 Agreement between expression estimates for replicates of Jurkat cells 27
Figure 2.5 Agreement between the expression level estimated computationally from RNA-seq data and those measured with an independent experimental method 30
Figure 3.1 Cell-type-dependent use of intronic PASs 51
Figure 3.2 Example and model to identify novel 3′-UTR isoforms 52
Figure 3.3 Evaluation of TECtool's performance 55
Figure 3.4 TECtool identifies previously unknown isoforms with cell-type-specific expression 57
Figure A.1 Overview of the study design 76
Figure A.2 Multithreading efficiency and running time / memory footprint trade-off 77
Figure A.3 Accuracy of transcript isoform abundance estimates inferred from in silico-generated sequencing data 78
Figure A.4 Comparison of different metrics for quantifying the accuracy of isoform abundance estimates 79
Figure A.5 Accuracy of gene expression estimates inferred from in silico-generated sequencing data 80
Figure A.6 Accuracy of 'present calls' 81
Figure A.7 Accuracy of expression estimates across all transcripts and genes 82
Figure A.8 Effect of 'native' short-read aligners 82
Figure A.9 Impact of bias correction settings on simulated data 83
Figure A.10 Expression level distributions across bins of transcripts and genes 83
Figure A.11 Cufflinks-based abundance estimates of single-exon transcripts 84
Figure A.12 Impact of gene structural features on expression estimates 84
Figure A.13 Agreement between expression level estimates for replicates of NIH/3T3 cells 85

Figure A.14 Replicate agreement between abundance estimates for features corresponding to expressed 3′ end processing sites 85
Figure A.15 Accuracy of 3′ end processing site abundance estimates inferred from Jurkat sequencing data 86
Figure A.16 Accuracy of 3′ end processing site abundance estimates inferred from NIH/3T3 sequencing data 87
Figure A.17 Impact of bias correction settings on abundance estimates from experimental data 87
Figure B.1 'Intronic' poly(A) sites are processed in a tissue-specific manner 90
Figure B.2 Sashimi plots of gene structures inferred from the RNA-seq data from different tissues 91
Figure B.3 TECtool analysis for bulk single-end or paired-end RNA-seq reads 92
Figure B.4 Overview of the region classification model in TECtool 93
Figure B.5 Features used in the model 94
Figure B.6 TECtool analysis flow for single cell data 95
Figure B.7 Evaluation of TECtool performance 96
Figure B.8 Distribution of expression levels inferred by Salmon 97
Figure B.9 Summary of RNA-seq samples from the protein atlas data set 98
Figure B.10 Update update update: Expression of TECtool identified transcripts across 32 human tissues 99
Figure B.11 TECtool identifies novel isoforms that are expressed in subsets of single cells 100
Figure B.12 TECtool analysis of an RNA-seq data set obtained from mouse CD4+ T cells 101

LIST OF TABLES

Table 2.1 Overview of surveyed methods 14
Table 2.2 Features and performance summary of the surveyed methods 36
Table 3.1 Datasets used for TECtool analysis 58

1 INTRODUCTION

The initial assembly of the human genome in the early 2000s [1] [2] paved the way for great advancements in both molecular and computational biology. A plethora of technologies and experimental designs emerged that enabled researchers to identify the sequence of thousands of organisms, discover new genes and proteins [3] and characterize single nucleotide polymorphisms (SNPs) or mutations [4]. Various Next Generation Sequencing (NGS) platforms [5] have not only been used in research, but also in clinical settings [6].

One of the fields that emerged from the progress in NGS is that of transcriptomics, which aims to describe the expression levels, the localization and the interactions of RNAs. The principal method in the field is RNA sequencing (RNA-seq) [7], which enables the capture and quantification of long RNAs such as mRNAs and lincRNAs, typically from populations of cells. Interestingly, taking advantage of by-products of endogenous enzymes, it has been found that RNA-seq can uncover RNA-RNA interactions [8] or gene fusions [9]. Specific protocols can be used to similarly characterize small non-coding RNAs such as miRNAs or snoRNAs. Yet other methods have been developed to map the sites of interaction of RNA Binding Proteins (RBPs) and miRNAs with their targets. These latter methods start with the freezing of *in vivo* interactions by ultraviolet light-mediated crosslinking, which is followed by the antibody-mediated immunoprecipitation the protein (with bound RNAs), hence the name of these methods (CLIP for crosslinking and immunoprecipitation) [10] [11] [12]. Much of the gene expression analysis can be carried out based only on sequencing of mRNA 5' or 3' ends. Cap Analysis Gene Expression (CAGE) [13] is used for identification and quantification of transcription start sites, whereas 3' end sequencing [14] [15] [16] [17] is used to identify and quantify terminal exon usage. Finally, high throughput methods like ribosome profiling [18] are available to study the translation of mRNAs.

The exponential growth [19] of heterogeneous transcriptomic sequencing data has created new demands to store, annotate, distribute and process these datasets. Public repositories for raw (for example SRA [20]) or processed (for example GEO [21]) have become important resources, to which the entire scientific community contributes. Specialized databases, providing detailed information about individual classes of RNAs like miRNAs [22] or snoRNAs [23] have also been developed, while meta-databases such as RNAcentral [24] combine and trying to improve the consistency of information generated elsewhere. One of the largest efforts to create a comprehensive catalog of transcripts and genes in different organisms is ENSEMBL [25], with the related GENCODE project [26], which aims to provide a

high-quality annotation of genes and transcripts for the human and mouse genome.

Given the tremendous growth of data, the demand for computational frameworks that enable researchers to handle datasets in an automated and reproducible way has also increased dramatically. General web-based platforms such as the Galaxy server [27] for biomedical research have gained many users. Our group developed and maintained the clipz web server [28], which enabled the automated, uniform analysis of data generated with a variety of different protocols such as small RNA-seq, mRNA-seq, 3′ end sequencing or crosslinking and immunoprecipitation (CLIP) of RNA-binding proteins. Command-line frameworks like Anduril [29], Snakemake [30] or specification languages like CWL [31] contribute to the effort of reproducible and open science [32].

1.1 COMPUTATIONAL ANALYSIS OF HIGH-THROUGHPUT SEQUENCING (HTS) DATA

Various best practices have been proposed for the analysis of HTS data [33]. Although as mentioned above, a variety of distinct protocols, developed for specific classes of RNAs, are available (for small RNA-seq, RNA-seq, 3′ end-seq, CLIP, etc.) some of the data analysis steps are relatively similar.

1.1.1 *Preprocessing*

Very frequently, samples are pooled (multiplexed) before sequencing to minimize the cost. To distinguish the samples, unique sequence identifiers are used when preparing individual libraries. These identifiers are used to separate (demultiplex) the samples again after sequencing. The results of the sequencing run are provided as FASTQ-formatted [34] files (Figure 1.1 A), generally one file per sample. Many statistics are calculated on the basis of these FASTQ files (Figure 1.1 A), such as the per base/read quality scores, per base/read G/C content, overrepresented or duplicated sequences. Tools like FASTQC [35] have been built to carry out these analyses. The initial statistics help identify low-quality reads or regions, that should be discarded or trimmed. Some methods such as TagDust2 [36] select 'mappable' part of each read. In TagDust2 this is done based on a user-provided Hidden Markov Model (HMM) [37] that describes the architecture of the reads and allows the recognition of primer dimers, barcode sequences, 5′ or 3′ adapters. Other tools like cutadapt [38] are also used to trim adapters (Figure 1.1 A) and perform quality filtering steps (Figure 1.1 A). When possible, PCR duplicates are removed. To avoid repeating the same operations unnecessarily many time, identical reads are generally collapsed using tools such as those from the Fastx-Toolkit [39] package. For downstream analyses, important information that needs to be extracted is the orientation of the reads, the average and standard deviation of the length of the fragments pre-

pared for sequencing. Finally, in many cases, it is helpful to convert the paired-end reads to single-end, and many methods like FLASH [40] are built to convert them efficiently.

1.1.2 *Alignment methods*

After the above-mentioned preprocessing steps have been completed, reads are mapped to reference sequences (Figure 1.1 A), which in most cases are the assembled genome of the studied species or the corresponding transcriptome. To ensure that efficient search, NGS alignment algorithms [41] follow a two-step procedure. In the first step a data structure like a hash table [42], a suffix tree [43] or a suffix array [44] is used to create an index of the reference sequences. In the second step, the actual mapping between reads and originating sequences is made. Outputs are currently written out in Sequence Alignment Map (SAM) [45] format or a compressed version of it like the Binary Alignment Map (BAM) [45] or the CRAM [46] format. Alignment files are written out either in 'random' order, or sorted by the name of the read or the coordinates of the read mapping in the reference sequence. For fastest lookup at later stages, the binary alignments should be indexed, and tools like samtools [45], BamTools [47], or Scramble [48] are built for sorting, indexing or performing other filtering tasks on alignment files. The nature of the genome and gene structure of the species of interest informs the downstream analysis of the alignments. For example, genomes have some degree of repetitiveness, as genes and genomic regions undergo duplication during evolution and therefore some reads map equally well to one or multiple loci. The multi-mapper rate depends also on the length of the sequenced reads, as short reads are more likely to match multiple genomic loci. Many studies ignore multi-mapping reads, although this leads, of course to mis-estimation of expression, as different genes have different repeat content and thereby arbitrary fractions of reads discarded. In some cases, such as for example for multi-copy small RNAs, discarding the multi-mapping reads would lead to entirely erroneous inferences of lack of expression of the multi-copy small RNA. An issue that has been challenging for a long time was that eukaryotic genes have introns, and thus reads that straddle splice junctions are at best difficult to map. Initially, strategies to handle this problem included generation of exon-exon junction databases to be added to the genome database for contiguous mapping, or hierarchical mapping to transcriptome and genome, both contiguously. The first fast short read aligners like bowtie [49] mapped the reads only contiguously, whereas second- and third-generation aligners like Tophat [50] or STAR [51] specialize in gapped alignments. Other aligners like segemehl [52] [53] are built for both purposes. Many of the spliced aligners can identify novel splice junctions, enriching the set of annotated junctions. Finally, more recently, the idea of pseudo-alignment was introduced [54] where for each read a list of compatible tran-

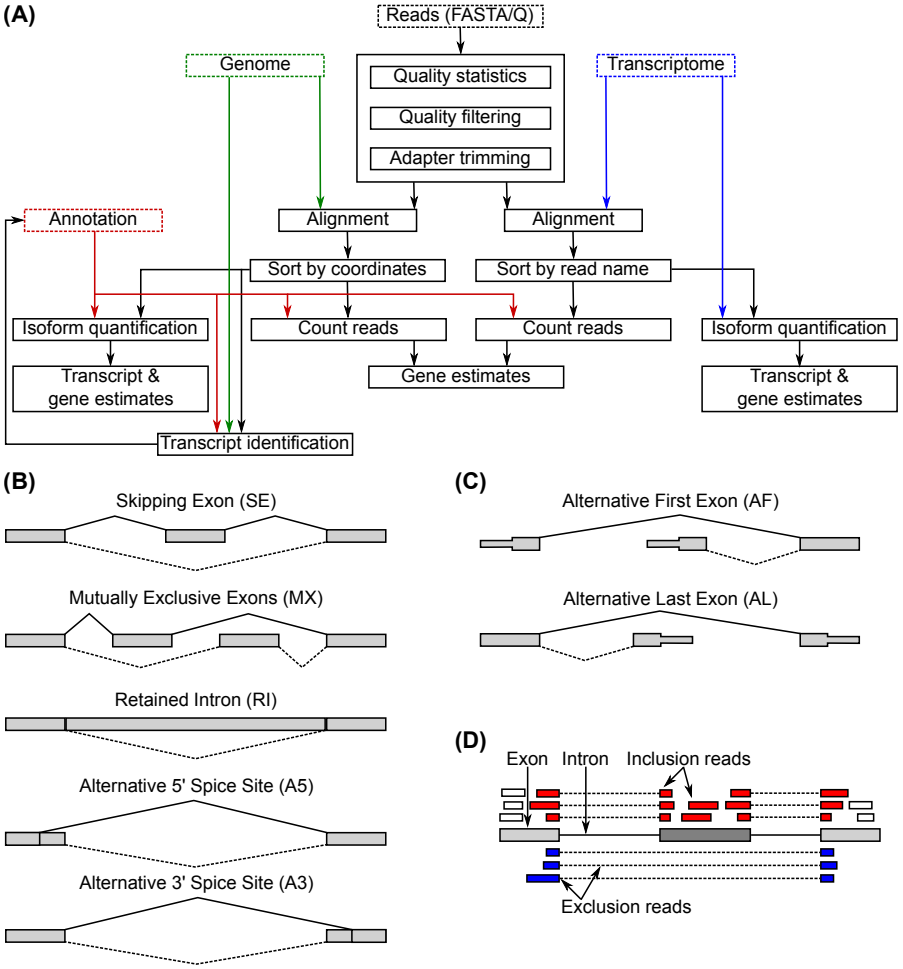


Figure 1.1: Analysis of HTS data. **(A)** Overview of the processing steps. Reads in FASTA or FASTQ format are preprocessed to generate statistics for data quality statistics, filter low-quality reads and trim adapters. Reads are then mapped to the genome (green) using an aligner that can handle spliced reads, or to the transcriptome (blue) with an aligner that only provides unspliced mappings. Genomic alignments are in general sorted by coordinates and indexed, while transcriptomic alignments are sorted by read name. Genomic or transcriptomic alignments are used to estimate the gene abundance using count-based methods given an annotation file (red). The same alignments are used as input for methods that estimate transcript abundance. Spliced alignments to the genome are used in methods that identify novel transcript isoforms. **(B)** Common splicing events: Skipped Exon (SE), Mutually Exclusive Exons (MX), Retained Introns (RI), Alternative 5′ Splice Site (A5), Alternative 3′ Splice Site (A3). Gray boxes indicate exons, lines and dashed lines indicate different transcript products. **(C)** Other RNA processing events. Same as in B, but showing Alternative First (AF) and Alternative Last (AL) Exons. **(D)** Reads considered for the calculation of the PSI score in count based methods. Light gray and dark gray boxes indicate exons, lines joining exons indicate introns. Red boxes are reads indicating the inclusion of the exon in the middle (dark gray), while blue reads indicate the exclusion of the read. Dashed lines indicate split reads between exons. White boxes correspond to reads that are not informative for whether the reference exon (dark gray) is included or not in a transcript.

scripts from which the read could originate is reported. This is done based on the word composition of the read and transcripts.

1.1.3 *Quantification methods*

One of the main uses of RNA-seq is estimating the expression of genes or transcripts (Figure 1.1 A). A naive way to infer the expression level of a gene from a genomic alignments of RNA-seq reads is to count reads that map to the union of exonic positions in the locus of that gene. That means that overlapping exons of a gene are merged into pseudo-exons and reads that overlap with these pseudo-exons will be counted towards the expression of gene. Typically, only uniquely mapping reads are used in this analysis. This crude approach has several disadvantages, which include that reads crossing real splice junctions are not considered and that the estimate of expression will be erroneous when the gene has multiple isoforms expressed at different levels. Other approaches consider known isoforms of genes and attempt to quantify their relative expression. After estimating the transcripts counts, the estimates are then aggregated at the gene level. In this analysis multi-mapping reads should be included. Many packages for counting reads have been developed over time, including HTSeq [55], featureCounts [56], bedtools [57] and QuasR [58]. They differ in the choice of programming language (python, R, bash) and in various choices they make for example in the treatment of multi-mappers. Their accuracy in quantifying expression is not entirely trivial to analyze as gold standard data sets are rare. This topic will be addressed also in chapter 2. Rounded counts per transcript/gene are generally used to infer differential expression across different conditions (e.g. health and disease), with the general assumption that the read counts associated with a transcript/gene follow the negative binomial distribution. The most used packages for differential expression analysis are edgeR [59] and DESeq [60]. For other analyses, measures such as Reads Per Kilobase per Million (RPKM) [61] or Transcripts Per Million (TPM) [62] are calculated.

The methods above were mainly developed to estimate gene expression levels. However, the throughput of sequencing methods became sufficiently high in recent years to allow one to investigate isoform expression, which can have strong impact on the cell. Moreover for recently discovered molecules such as lncRNAs, the annotation may be highly incomplete and thus the deep sequencing data can also serve to improve the annotation of gene structures. Different computational approaches have been designed for estimate relative abundance at transcript level. They take as input spliced alignments to the genome or contiguous alignments to transcripts. A challenge that these methods need to address is handling reads that map to multiple transcripts, that can either come from the same locus, or from different loci. Most methods use the Expectation Maximization (EM) approach [63]. During the Expectation (E) step reads are proportionally assigned to transcripts according to the isoform abundance, whereas

during the Maximization step (M) the transcript isoform abundances are recalculated from the previously assigned read counts [64] [65]. Additional characteristics such as the availability of paired end data [64] or of information about positional or sequencing biases [64] are also incorporated in the inference. Although the models underlying the different methods are generally similar, small differences in parameters or even in the implementation of the algorithm can lead to different results. Evaluating the performance of these methods is not an easy task, especially in the absence of 'ground truth' isoform abundance. All methods rely to some extent on simulated data, which likely lacks biases introduced by difficult to control experimental factors. In chapter 2 we evaluated the performance of some the most widely used methods using simulated and real human and mouse datasets.

1.1.4 *Transcript identification methods*

The above sections describe analyses that are done with RNA-seq data assuming that the set of expressed transcripts is known. However, one of the main applications of RNA-seq is the identification of novel transcripts/isoforms [66] [67]. There are two types of tools that try to solve this problem, one using only alignments of reads to genome and building chains of exons, the other using both genomic alignments as well as available transcript annotation, to generate an enriched annotation. The first approach is used for organisms where the genomic sequence was recently determined and thereby the gene annotation is very limited, whereas the second approach is more common for well-studied organisms like human or mouse, where the interest is frequently in identifying novel transcripts that have been missed so far, probably because of their highly specific expression. Although different tools are available [66] [67] they share the same main principle [65]. Namely, alternative transcript structures are modeled with a splice graph, different isoforms corresponding to different paths through the graph [68]. The nodes of the graph represent exons whereas directed arrows represent splicing events. All possible traversals of the graph, corresponding to potential transcripts, can be enumerated, using for example a breadth-first-search (BFS) algorithm. Expectation-maximization (EM) is then used to estimate the relative transcript abundance. While the available transcript reconstruction methods can identify reasonably well internal exons, which are supported by spliced reads at both ends, they have relatively poor performance in identifying the transcript ends, that are not as sharply defined as the splice junctions and have a poorer coverage due to the loss of short reads generated from the ends during the fragmentation step of RNA-seq sample preparation. In one of my projects I have co-developed a new method (chapter 3) for improved identification of novel terminal exon isoforms.

1.1.5 *Analysis of alternative splicing isoforms*

Splicing [69] [70] [71] is the process where pre-mRNAs are converted to mature mRNAs by the removal of 'intronic' sequences. This process takes place co- or post-transcriptionally [72], exons being joined together to form the mature transcripts, while the introns are degraded through specific mechanism or further processed to generate other functional molecules such as small nucleolar RNAs or microRNAs [73]. In higher eukaryotes, many exons are not constitutively spliced into the transcript, but rather their inclusion or exclusion depends on the binding of splicing factors in regions around the splice sites. This leads to 'alternative splicing' (AS) and to the production of different transcript isoforms for individual genes, in different cell types or conditions. Virtually all (more than 95%) of multi-exon human genes give rise to alternatively spliced isoforms [74].

There are again various approaches to quantify alternative splicing (AS) events or isoforms computationally [75], two being most common. The first approach is exon-centric: expression of each exon of a gene is quantified across multiple samples, and the level of inclusion/exclusion of a given exon is determined relative to the average over other exons of that gene. The second approach is transcript based: one attempts to quantify the expression levels of alternative isoforms, deriving from these inclusion rates of individual exons. A generally adopted measure of exon usage is the Percent Splice In (PSI) score [76] which indicates what fraction of the transcripts of a gene contain a specific exon, from among all the transcripts that cover the genomic locus of that exon. This measure is generalized to various splicing events (Figure 1.1 B and 1.1 C) such as exon skipping (SE), mutually exclusive exons (MX), intron retention (RI), use of alternative 5' splice sites (A5), alternative 3' splice sites (A3) alternative first (AF) or alternative last (AL) exons. Tools like MISO [77], MATs [78] or rMATs [79] use spliced alignments of reads to the genome (Figure 1.1 D) in order to calculate the PSI score for each exon. MISO can be used for alternative splicing analysis of a single sample or for the comparison of two samples, MATS is used for the comparison of two samples and rMATs for the comparison of samples with replicates. Other tools like SUPPA [80] use as input the transcripts abundance estimates (determined by transcript isoform quantification methods described above and further discussed in chapter 2) and calculate the PSI score of each exon by calculating the fraction of the transcripts that include it over the transcripts that either include or exclude it. One of the most common forms of alternative 3' splice sites is the tandem acceptor sites (at the NAGNAG pattern) [81]. Because these variants differ by very few (often 3) nucleotides, special care is needed to make sure they are not treated as errors in sequencing or alignment. Similar considerations apply to a recently described type of exons, called micro-exons, that are especially abundant in neural tissues. Specialized databases containing these exons are now available [82]. There are other, more complex splicing patterns that are also of interest in specific studies [83]. In many cases there are events that are ob-

served in specific tissues or health conditions and this information is missing from existing annotation databases. For this purpose specialized databases have been constructed [84] that contain pre estimated inclusion scores for different tissues and organisms. Finally, specialized tools for the quantification of specific splice variations such as intron retention have also been developed [85] [86] [87] [88].

1.1.6 *Alternative polyadenylation analysis*

Besides splicing, eukaryotic mRNAs are also modified by the addition of a cap at the 5′ end [89] and a stretch of poly(A)s at the 3′ end [89]. The position where transcription is initiated varies, leading to alternative TSS [90], and similarly, where the poly(A) tail is added is also not invariant, leading to alternative terminal exons. This depends on the RBPs [91] attached to the region and the availability of poly(A) sites [92]. Recently, it has been found that some transcripts differ only in the length of their 3′ untranslated regions (3′ UTRs), one isoform carrying a long and the other a short version of the 3′ UTR [93]. Identifying the used poly(A) site (distal or proximal site) is important because 3′ UTRs interact with many RNA-binding proteins that determine the traffic, localization stability and translation rate of the mRNA. Thus, many protocols for sequencing mRNA 3′ ends have been developed [14] [15] [16] [17]. Corresponding computational pipelines were developed [94] [95] and databases containing the identified poly(A) sites have been constructed [96] [97] [98] [94]. However, some efforts have been initiated to use the vast amount of RNA-seq data already generated to infer terminal exon isoforms expression. The usage of alternative promoters (AF) or alternative terminal exons (AL) (Figure 1.1 C) can be quantified with tools initially built to study alternative splicing (exon-based or transcript-based approaches discussed previously). Additionally, RNA-seq based algorithms have been developed [99] [100] [101] to study the differences in alternative terminal exons usage. In chapter 2 and chapter 3, we used poly(A) sites identified from 3′ end sequencing protocols to evaluate the performance of transcript quantification algorithms and identify novel terminal exons respectively.

1.1.7 *Long non-coding RNAs*

Deep sequencing has also enabled the discovery and further characterization of many different types of non-coding RNAs. Based on their length, these are categorized as small non coding RNAs, that are 20-30 nucleotides long (miRNAs, piRNAs, and siRNAs), intermediate length non-coding RNAs of 50 to 150 nucleotides (snoRNAs, tRNAs, and U snRNAs) and long non-coding RNAs (lncRNAs) that are 200 bases or longer. LncRNAs can originate from intronic regions of protein-coding genes, from intergenic regions of protein-coding or non-coding genes (lincRNAs) or antisense regions of protein-coding genes (antisense lncRNAs) [102]. LncRNAs may or may not contain

poly(A) tails [103] and can be detected either in the nucleus or/and in the cytoplasm [104].

Various methods have been used for the identification of lncRNAs [105] like 3′ end sequencing for transcript ends and H3K4me3 chromatin maps for the identification of transcription start sites [105]. RNA-seq (mainly total RNA-seq) is used for the identification of the structure of transcripts [106] and the estimation of their expression. LncRNAs are in general lowly expressed [107], so it is important to use quantification methods that are accurate in this expression regime. Some parts of chapter 2 discuss this issue. Transcript reconstruction algorithms are also crucial for the identification of novel lncRNAs since this field is relatively new and the set of lncRNAs in databases is incomplete. Efforts to catalog lncRNAs are described in references [108] [109] [110]. Chapter 3 discusses transcript reconstruction methods that we developed in our group to identify novel transcripts in lincRNA regions. An important step apart from the identification is the characterization of the potential novel lncRNA. Ribosome profiling provides a good indication of whether a lncRNA can be translated or not. Computational methods have been developed that use machine learning approaches to predict if a newly identified transcript can be classified as lncRNA [111].

1.1.8 *Visualization*

Visualization is an important part for interpretation of HTS data. Multiple genome browsers have been developed over the years. Many of them can be used over the web like the UCSC genome browser [112], while others are used locally like IGV [113]. R libraries [114] are commonly used to visualize genomic tracks. Finally, Sashimi plots [115] are useful to visualize the splice junctions that are used.

2

COMPARATIVE ASSESSMENT OF METHODS FOR THE COMPUTATIONAL INFERENCE OF TRANSCRIPT ISOFORM ABUNDANCE FROM RNA-SEQ DATA

2.1 ABSTRACT

2.1.0.1 *Background*

A detailed understanding of the regulation of gene expression, including transcription start site usage, alternative splicing and polyadenylation, requires accurate quantification of gene expression levels down to the level of individual transcript isoforms. To this end, a variety of methods for estimating transcript isoform abundance from RNA sequencing data have been proposed. To comparatively evaluate their accuracy, we have used both synthetic data as well as an independent experimental method for quantifying the abundance of transcript ends at the genome-wide level.

2.1.0.2 *Results*

We found that many tools have good accuracy and that they yield better estimates of gene-level expression than commonly used "count-based" approaches. Transcript or gene-level features such as nucleotide composition and intron/exon structure appear to have little influence on the accuracy of expression estimates, which correlates most strongly with transcript/gene expression levels. Finally, we found large differences in the memory and runtime requirements of the different tools, factors that are likely to be important in their adoption by the user community.

2.1.0.3 *Conclusions*

As many methods for quantifying isoform abundance with comparable accuracy are available, a user's choice will likely be determined by factors such as the memory and runtime requirements, as well as the availability of methods for downstream analyses. Sequencing-based methods to quantify the abundance of specific transcript regions could complement validation schemes based on synthetic data and quantitative PCR in future or ongoing assessments of RNA-seq analysis methods.

2.2 BACKGROUND

The general availability of high-throughput sequencing technologies greatly facilitated the detection and quantification of RNA species,

including protein-coding RNAs, long non-coding RNAs, and microRNAs, in many different systems. In higher eukaryotes, the vast majority of protein-coding genes express multiple transcript isoforms [116] [117] [118]. Although a substantial proportion of transcript isoforms may result from stochasticity in the splicing process [81] [119], striking examples of isoform switching with large impact on cellular phenotypes are also known (for example, [120] [121]). Tissue-specific splicing patterns have been linked to the expression of specific RNA-binding proteins [122], some of which appear to act as 'master' regulators of alternative splicing in individual tissues [123]. For example, muscleblind-like proteins 1 and 2 (MBNL1/MBNL2) are expressed in mesenchymal cells and their downregulation facilitates somatic cell reprogramming [124], while the epithelial splicing regulatory proteins 1 and 2 (ESRP1/ESRP2) establish epithelia-specific patterns of isoform expression [125]. Nevertheless, despite the long history of the field, the functional relevance of most isoforms that can be detected with sequencing approaches remains unclear [126], particularly in light of the rapid change of isoform usage pattern in evolution that indicates relatively weak selection pressure [127].

Analysis of expression pattern is often one of the first steps towards understanding a gene's function. However, transcript isoform abundance is almost always quantified indirectly; most of the sequencing technologies that are currently used yield reads that are short ($\leqslant$200 nt) relative to the length of eukaryotic transcripts (2.2 kb in mammals, on average) [128] and thus, a sequenced read can typically be assigned to more than one isoform. This is not the case with the technology developed by Pacific Biosciences that enables sequencing of full-length cDNAs [129]. A drawback of this technology is, however, that the throughput is relatively low, of the order of 104 transcripts, which does not allow accurate quantification of transcript abundance. Furthermore, the error rates are relatively high, making the transcript identification non-trivial. Thus, accurate and cost-effective quantification of the complete repertoire of full-length expressed transcripts, which are in the range of hundreds of thousands per cell [130], remains an open problem.

As RNA sequencing (RNA-seq) has become commonplace in molecular biology laboratories, a variety of computational approaches has been proposed for isoform reconstruction from short read sequencing data (see, for example, [66]). Similarly, quite a number of methods has been developed for the inference of isoform abundance (reviewed in [75]). While short read alignment and transcript reconstruction methods have been extensively benchmarked recently [66] [131], only one study, rather limited in scope, evaluated some isoform quantification methods [132]. Independently and comprehensively evaluating the accuracy of such computational methods is difficult, because experimental validation strategies by, for example, quantitative PCR are typically restricted to just a limited number of isoforms (see, for example, [133]). Developers therefore typically evaluate their tools on synthetically generated datasets which may not capture adequately the complexities of RNA-seq experiments.

In this study we carried out a systematic evaluation of a large number of methods for isoform quantification from RNA-seq data. We used not only synthetic, but also genome-wide experimental datasets. We took advantage of newly developed protocols for quantifying the abundance of distinct RNA 3′ ends, which result from the use of alternative 3′ end processing sites. These protocols allow a comprehensive surveillance of 3′ end processing site usage, with a method that is distinct from RNA-seq [14] [134] [16]. From two types of cells and from two species (human Jurkat T cells or mouse NIH/3T3 cells) we prepared two libraries, one with an RNA-seq protocol and the other with a protocol for capturing the 3′ ends of polyadenylated RNAs. We submitted the aligned RNA-seq reads to the entire panel of computational methods for estimation of transcript isoform abundance. We then compared these estimates with those that we obtained independently, through the analysis of the corresponding 3′ end sequencing data.

Our results indicate that many of the available methods have comparable accuracy, and that the abundance of highly expressed isoforms is more accurately inferred than the abundance of isoforms with low expression levels. We further found that even the quantification of gene expression is more accurate when gene expression levels are computed by cumulating the levels of transcript isoforms than when ignoring the transcript structures. Given that many methods are available that differ little in accuracy, a user's choice will likely be determined by factors such as the memory and runtime requirements, as well as the availability of methods for downstream analyses such as differential gene/transcript expression.

2.3 RESULTS

We initially performed an extensive literature survey to identify tools that were developed for inferring the abundance of transcript isoforms from RNA-seq data. Although we tried to include as many of these as possible, our study setup required that tools are able to quantify a set of transcripts that we provided as input, thereby separating the problem of transcript reconstruction from that of abundance quantification. To be able to interpret the results, we further focused on methods that have been duly described in the literature. Lastly, we thought that ease of use would be critical for the adoption of the tool by the user community and we did not pursue methods which we were unable to implement within a reasonable amount of time. Table 2.1 lists the remaining 11 tools, together with their underlying principle, input requirements, and references. A description of how each of the tools was applied is provided in the Methods section.

2.3.1 *Runtime and memory requirements differ substantially between tools*

Most of the tools that we surveyed have previously been tested by the developers on simulated data. Here, we have used the Flux Sim-

Table 2.1: Overview of surveyed methods. The columns are: method name, sequences to which reads are compared (transcripts or genome), principle of the method, year of release, and associated reference(s)

Name	Reference sequence[1]	Principle	Released
BitSeq	Transcripts	Bayesian estimation of parameters of a model that explains the read-to-transcript alignment data. Reads are assumed to be sampled independently, without positional bias from transcripts, such that the probability of an alignment starting at a given position of a transcript is inversely proportional to the transcript length. Sub-optimal alignments are used to estimate the 'background' of spurious alignments.	2012 [135] [136]
CEM	Genome	Component elimination expectation-maximization approach to estimating the parameters of isoform abundance. For each gene it aims to find a 'sparse' solution, with few expressed isoforms. Read sampling from isoforms is assumed to obey a quasi-multinomial distribution, in which positional and other biases are modeled as an effective distribution which could be, for example, uniform (no positional bias) or exponential (modeling the process of RNA degradation).	2012 [137]
Cufflinks	Genome	Bayesian approach to estimating transcript abundances by explicitly modeling the length of the fragments expected from RNA-seq. It assumes that for a given gene, reads are sampled independently with uniform probability along transcripts and in proportion to the transcript abundance between transcripts. Thus, if a read can be assigned to two transcripts of different lengths, the transcript with a shorter effective length will have a higher probability of giving rise to the read.	2010 [138]
eXpress	Transcripts	Similar to Cufflinks, but it includes modeling of errors and indels and it has a different model for fragment length selection. Unlike Cufflinks and most other methods, eXpress processes read alignments 'on-line' so that it can be integrated into real-time analysis pipelines.	2012 [139]
IsoEM	Genome	Expectation-maximization approach to inferring isoform abundances that are consistent with the coverage of isoforms by reads. The coverage is assumed to be uniform along an isoform. Base quality scores are taken into account in computing the probabilities of alignments. In the E-step, the expected number of reads derived from a given isoform is computed and in the M-step, the relative frequencies of isoforms are estimated.	2011 [140]
MMSeq	Transcripts	Models the read data as Poisson-distributed variables with rates that depend on the abundance of the regions of the transcripts with which the reads are compatible and on the sequence-dependent bias in capturing the sequences. Priors on transcript abundances are Gamma-distributed. Sequencing errors are not modeled, there is only a filter on the minimal quality of considered alignments.	2011 [141]
RSEM	Transcripts	Models the probability of observing a read as the sum of the relative abundance of the transcript to which the reads maps times the probability of the read mapping to the transcript, and infers transcript abundances by expectation maximization.	2009 [142] [143]
rSeq	Transcripts	Models read data as Poisson-distributed variables with rates that depend on the abundance of the regions of the transcripts with which the reads are compatible.	2009 [144]
Sailfish[2]	Transcripts	Expectation-maximization method for explaining the abundance of k-mers inferred from the reads in terms on the abundance of the transcripts with the associated k-mer abundances.	2014 [145]
Scripture	Genome	Transcript abundance is calculated as reads per kilobase of exonic sequence per million aligned reads, given the alignments of the reads to the genome and the annotated/reconstructed transcript.	2010 [106]
TIGAR2	Transcripts	Models the read data in terms of a large number of parameters which include, beyond the relative abundance of the transcripts, the read length distribution, the nucleotides, and alignment state and quality at the first and second position of the read.	2013 [146] [147]

ulator software [148] to generate reads corresponding to GENCODE-annotated transcripts Supplementary Figure A.1). To assess how the runtime complexity, memory requirements, and accuracy of the different programs depended on the sequencing depth we generated sets of 1, 3, 10, 30, and 100 million single-end reads, the latter two values being in the range that is currently obtained from sequencing a typical RNA-seq library on broadly used next-generation sequencing platforms. We found that the tested programs differ substantially in their runtimes and memory footprints, as measured under defined conditions on a dedicated machine (maximum available memory = 64 Gb). As shown in Figure 2.1, the CPU times necessary to process the different datasets span about three orders of magnitude when a single processor is used (Figure 2.1 A), and two orders of magnitude when the multi-threading option (16 cores; Figure 2.1 B) is used. In particular, the times required to process the alignments of 100 million in silico-generated reads range between approximately 7 min (IsoEM) and more than 1 week (TIGAR2) when a single processor is used, and between about 5 min (IsoEM) and 8 h (RSEM) when 16 cores are available for the tools that support multi-threading (TIGAR2 does not). With the exception of Sailfish, runtimes strictly increased with the number of processed read alignments. Assuming that a method-specific, but largely sample size-independent time span is required to index the supplied transcriptome, time complexities for most of the quantification algorithms appear to be approximately linear. Sailfish's runtimes seem to be the highest for the smallest dataset, presumably because the convergence of estimation is slow for small datasets, when the vast majority of transcripts are sparsely covered. Notably, Sailfish computes abundances based on raw read sequences rather than alignments. Thus, whenever alignments are dispensable, a considerable amount of time (typically 1 h or more) can be saved on sample pre-processing compared to all other methods (refer to [131] [149], [150] for an overview of 'mapping' times for some short-read aligners and conditions). Enabling multithreading had only a limited impact on runtimes (Supplementary Figure A.2 A), with several of the tools hardly benefiting at all (maximum ratio between runtimes at 1 and 16 cores approximately two-fold or less for CEM, eXpress, MMSEQ, rSeq, and Scripture). However, RSEM (approximately 5.9-fold speedup for 30 million reads) and BitSeq (approximately 4.2-fold speedup for 100 million reads), two methods with the highest single-processor running times had the highest speedup when multiple processors were provided. Memory footprints also spanned almost two orders of magnitude between tools, both when using a single or multiple cores (Figure 2.1 C, D). For approximately half of the tools (CEM, eXpress, MMSEQ, Sailfish, Scripture, TIGAR2) the memory footprint seems to be largely independent of the sample size. For the remaining tools (BitSeq, Cufflinks, IsoEM, RSEM, rSeq) the memory footprint increases with the sample size. Although IsoEM seems to trade off a relatively large memory footprint (from <10 to >30 GB) for extremely short running times, we did not observe a general inverse correlation between the running time and memory usage of individual methods

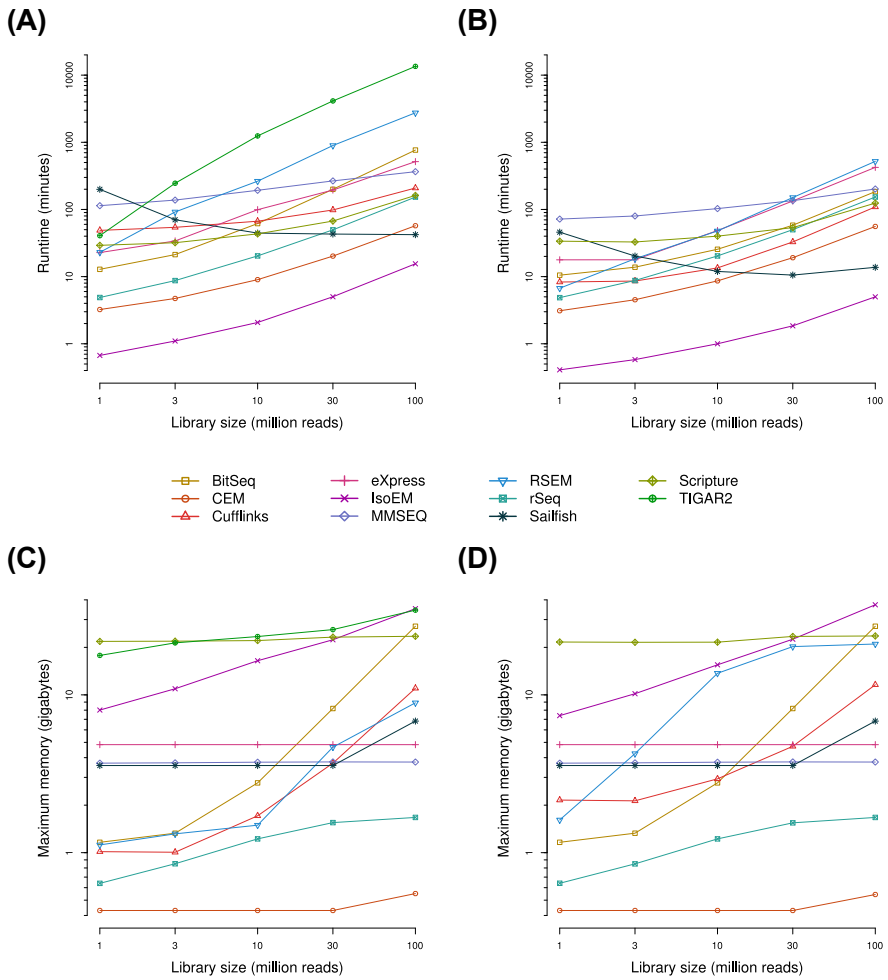


Figure 2.1: Running time and memory requirements. Transcript isoform abundances were estimated with each of the indicated methods from in silico-generated datasets of different 'sequencing' depths. The running times **(A and B)** and memory footprints **(C and D)** are shown as a function of sequencing depth. Programs were run on either one (A and C) or 16 cores (B and D). Note that TIGAR2 is missing in (B) and (D), because the method does not support the use of multiple cores

(rs = 0.13 and -0.13 at 100 million reads for 1 and 16 cores, respectively) (Supplementary Figure A.2 B, C).

2.3.2 *Most methods infer transcript abundances with good accuracy even from sparse datasets*

Our main objective was to evaluate the accuracy of isoform expression estimates produced by various methods. Consistent with current expectations about the number of expressed transcripts in a given cell type, the read simulation software only assigned non-zero expression to approximately 10.2% of all transcripts supplied to it as input (19'004 out of 187'176). To avoid the situation that our results are dominated by how different methods handle transcripts that are essentially not expressed, we initially restricted our initial analysis to the set of expressed transcripts. These were those for which the sim-

ulation software assumed non-zero expression values. When comparing the abundances of these transcripts as inferred by each method with the 'ground truth' (Figure 2.2 A and Supplementary Figure A.3), we found that nine out of 11 programs exhibit very good performance (Spearman correlation coefficient rs >0.9 for $\geqslant$107 reads). As expected, correlations generally improved with increasing library sizes, in a monotonic fashion and asymptotically towards saturation. For most methods, estimation accuracies reached a plateau at or around a read depth of 30 million reads, indicating that further increases in read depth are unlikely to significantly improve their results. In particular, Spearman correlation coefficients peaked at above 0.95 for six of the methods (BitSeq, eXpress, IsoEM, RSEM, Sailfish, and TIGAR2) and above 0.9 for a further three methods (CEM, MMSEQ, rSeq). Both Cufflinks and Scripture performed considerably worse than all other methods, with the corresponding correlation coefficients barely surpassing 0.75. The influence of the library size on accuracy varied somewhat between methods, with the total gain from the sparsest to the richest dataset ranging from approximately 0.01 (Cufflinks) to approximately 0.08 (BitSeq). Out of the nine most accurate methods, MMSEQ appears to be the least sensitive to the influence of read depth (approximately 0.04 gain in accuracy). In order to rule out that our chosen metric for measuring accuracy is prone to producing idiosyncratic results, we have compared it with both the Pearson correlation coefficient and the root mean square error (Supplementary Figure A.4 A). The relative performance of the methods changed only little, indicating that the results were robust with respect to the metric that we chose. Thus, with few exceptions, all methods produce highly accurate transcripts isoform abundance estimates even at moderate read depths.

2.3.3 *Explicit modeling of transcript isoforms leads to more accurate estimation of gene expression levels than count-based methods*

Gene expression levels are typically derived from RNA-seq-based data by intersecting the genome coordinates of 'uniquely-mapped' reads with the loci of annotated genes and taking into account the length of the transcript that is expressed from a given locus. As may be immediately apparent, this procedure has several limitations. The first is that it is generally unclear what transcript to consider for each locus, when correcting for transcript length. What is typically used is the total length of the 'union exons', which is clearly incorrect when the gene expresses multiple isoforms with different relative abundances and different sequences of exons. A second drawback is that the proportion of reads that are discarded depends on the repeat content of the gene with an unknown impact on the accuracy of gene expression estimates. Finally, reads that map across splice boundaries and are informative particularly for estimating the expression of individual isoforms, may be discarded by the simple counting procedure. This problem will preferentially affect expression estimates for genes

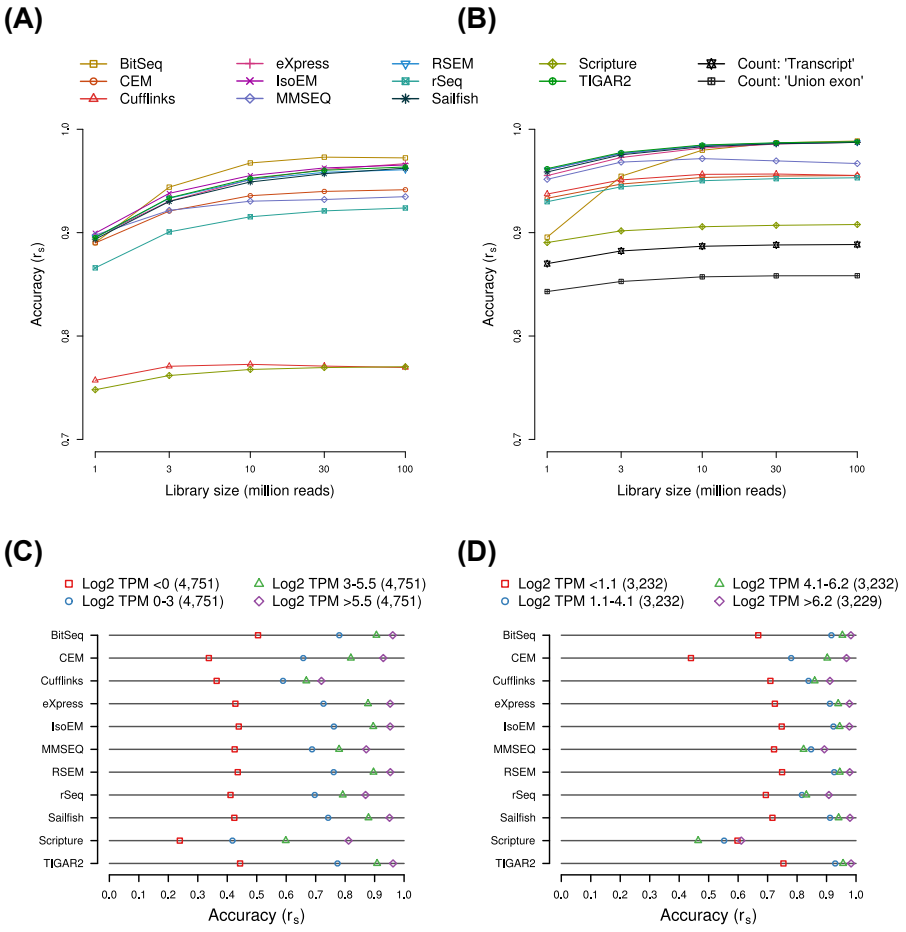


Figure 2.2: Influence of sequencing depth and expression levels on the accuracy of expression estimates. Transcript isoform and gene expression levels were estimated with each of the indicated methods from in silico-generated datasets of different 'sequencing' depths. The accuracy of a method was assessed in terms of the Spearman correlation coefficient (rs) between the estimates and the known input levels ('ground truth') of expressed transcripts **(A)** and genes **(B)**. Based on their true abundances, transcripts **(C)** and genes **(D)** were distributed across four bins of expression levels. Estimation accuracies as in (A) and (B) are indicated for each method and bin. The numbers of transcripts and genes in each bin are indicated together with the expression ranges that they cover. Estimates are based on a sequencing depth of 30 million reads

with a large number of exons and isoforms. Thus, one expects that even gene-level estimates of abundance are improved by the appropriate treatment of transcript isoforms. To test how accurately gene expression levels could be estimated by the benchmarked methods compared to count-based methods, we implemented two variants of count-based gene expression level estimation ('union exon' and 'transcript'-based counting, see Methods). The first method is both simple and widely used, but it has the pitfalls mentioned above. The second method tries to correct some of the inaccuracies of the simple union exon counting method by taking multi-mappers into account and avoiding artificial gene structures. If a method provided gene-level estimates (as is the case for Cufflinks, IsoEM, MMSEQ, RSEM, and rSeq) by default we used these values, otherwise we aggregated estimates of transcript abundances to obtain such estimates. We then compared these gene expression estimates to the true gene expression levels, which were also derived by aggregating the known isoform abundances. When considering only the 12'925 expressed genes (log2 TPM > -5; approximately 26.5% of all genes), the results (Figure 2.2 B and Supplementary Figure A.5) were qualitatively very similar to those that we obtained at the level of transcript expression (Figure 2.2 A and Supplementary Figure A.3): estimates of gene expression levels that were produced by or derived from the output of most methods are quite accurate and the accuracy increases with sequencing depth towards saturation. Only BitSeq's gene-level estimates were strongly sensitive to the size of the input library, in the range of approximately 0.90 for 1 million reads to approximately 0.99 for 30 million reads or more. The same six methods that yielded the most accurate transcript abundances (BitSeq, eXpress, IsoEM, RSEM, Sailfish, and TIGAR2) gave the most accurate gene level expression estimates: all achieved peak Spearman correlation coefficients of 0.98 or higher. CEM, Cufflinks, MMSEQ, and rSeq reached Spearman correlation coefficients of at least 0.95. Scripture, when provided with more than 1 million reads, was also able to estimate gene expression with good (rs >0.9) accuracy. In contrast, the count-based methods only achieved moderate accuracy (maximum rs = 0.89 and rs = 0.86 for the 'union exon' and 'transcript' methods). As suggested by the scatter plots in Supplementary Figure A.5, the limited accuracy of either method is largely due to the underestimation of true expression and, as expected, this short-coming is more pronounced in the 'union exon' method. As with the transcript estimates, choosing another metric has little impact on the overall ranking/presentation of results (Supplementary Figure A.4 B). Taken together, these results clearly demonstrate that although the accuracy of count-based methods may perhaps benefit from more elaborate procedures for addressing ambiguities in the assignment of reads to loci and transcripts, they still fall short of methods that probabilistically model the generation of RNA-seq data, taking into account transcript isoforms and the sampling of reads from transcripts.

2.3.4 *High expression levels are more accurately estimated than low expression levels*

Higher transcript coverage by reads is expected to increase the accuracy with which transcript abundance is estimated. The coverage depends on both the depth of sequencing as well as on the transcript abundance, and indeed we found that the size of the read library has a positive influence on the accuracy of expression estimates. To evaluate the extent to which 'true' abundance influences the accuracy of transcript abundance estimates, we grouped both expressed transcripts and genes by their 'ground truth' expression into four equally sized bins: low (log2 TPM <0 or 1.1), medium-low (0 or 1.1 < log2 TPM <3 or 4.1), medium-high (3 or 4.1 < log2 TPM <5.5 or 6.2) and high abundance (log2 TPM >5.5 or 6.2), with the first and second numbers referring to the ranges for transcripts and genes, respectively. The overall ranking of tools in terms of their accuracy within expression level bins (Figure 2.2 C, D) largely reflects what we observed when evaluating the performance on expressed transcripts or genes (Figure 2.2 A, B). However, the accuracy of transcript expression level estimates degrades progressively from high to low expressed transcripts, with the most drastic drop between the medium-low and low (less than one transcript in 1 million transcripts) abundance (correlation coefficients for the most accurate tools change from approximately 0.75 to approximately 0.4/0.5, at 30 million reads, Figure 2.2 C). Similarly, estimation accuracies on the gene level differ little across the three bins of most highly expressed genes (mean rs = approximately 0.92, 0.87, 0.85 for the 'high', 'medium-high', and 'medium-low' bins, respectively), but drop most strongly for the bin with the least expressed genes (mean rs = approximately 0.68). Thus, our analysis confirms the expectation that low abundance and, consequently, sparse transcript coverage leads to noisier estimates of expression. However, for genes whose expression levels are in the top three quartiles, the estimates provided by the tools agree very well with the 'true' expression levels.

Because different methods appear to handle quite differently transcripts with very low abundance, we sought to further investigate their accuracy in this expression range in particular. More specifically, we determined the rates at which: (1) transcripts or genes that are not expressed are estimated to have non-zero expression (false positive rate); and (2) transcripts or genes that are expressed and are also inferred by a tool to have non-zero expression levels (true positive rate). It should be noted that when dealing with real rather than synthetic datasets, one does not know whether a specific transcript truly had a copy number of 0 in the sample or not. When no evidence of expression is found, some of the Bayesian methods (BitSeq and MMSEQ) strictly assign non-zero 'prior' expression probabilities to transcripts, and thus they do not, strictly speaking, produce any 'false negatives'. Nevertheless, even for these methods it may be relevant to determine how well very limited evidence of expression is handled, and whether transcripts with no such evidence really get assigned 'prior'

expression values. Thus, after consulting the developers, we have assigned transcripts with an expression estimate which was essentially the method-specific prior value an estimate of zero (see Methods), and then determined the false and true positive rates of all methods. In general, we found that the surveyed methods vary quite considerably in their ability to make accurate 'present calls' for transcripts and genes and that tools that exhibit low false positive rates tend to falsely assign zero estimates to a higher fraction of transcripts or genes, as expected (Suppementary Figure A.6). In this category are IsoEM, RSEM, rSeq, Sailfish, TIGAR2, MMSEQ ('prior' expression levels handled as described above), as well as Cufflinks and Scripture (the latter two only when considering gene level estimates). In contrast, CEM, eXpress, BitSeq (zeroed 'priors' as described above), Cufflinks, and Scripture (on the level of transcripts), and, in an extreme manner, the unmodified estimates from BitSeq and MMSEQ show the exact opposite behavior. As expected, the rate of true positive calls increases with increasing read depth, as does the rate of false positives. The increase in true positive calls is particularly apparent for lowly expressed genes and transcripts, for which the true positive rate increases steeply up to 30 million reads (Supplementary Figure A.6 E, F). Overall, deeper datasets yield an increased fidelity of making present calls. Consistent with these results, the Spearman correlation coefficients, when calculated across all transcripts and genes (Supplementary Figure A.7 A, B), are considerably lower than when only expressed features are considered (Figure 2.2 A, B). Given that most of the annotated transcripts were considered 'not expressed' in our synthetic dataset, the tools that trade off specificity for sensitivity (BitSeq, CEM, eXpress, MMSEQ) were most affected by the inclusion of not expressed transcripts. Taken together, these analyses indicate that the amount of starting material, the features of interest, and the obtained read depth are all among the factors that influence the accuracy of expression estimates and may play a role in the choice of the method that should ultimately be used for data analysis. Nevertheless, moderate sequencing depth of a few tens of million reads seems to be sufficient for an accurate estimation of most except the very lowly expressed transcripts by many of the available methods.

2.3.5 *The alignment program and bias correction options have little impact on the accuracy of abundance estimates*

Some of the surveyed methods strongly recommend the use of a specific short-read alignment program. By default, RSEM even calls such an aligner (Bowtie) internally. Thus, we asked whether the choice of alignment program impacts the accuracy of isoform abundance estimates that are produced by these methods. Surprisingly, we found that the aligner has a relatively small impact on estimation accuracy, regardless of whether one considers transcripts or genes, and only expressed or all features (Supplementary Figure A.8). If anything, with the exception of CEM, all methods performed better when sup-

plied with read alignments prepared with our custom pipeline that employs the segemehl aligner than when alignments produced by either Bowtie1 (MMSEQ, RSEM) or TopHat2 (Cufflinks, Scripture) were provided. RSEM had the highest gain in accuracy, around r_s = 0.05 or r_s = 0.03 on the transcript- and gene-level, respectively. On the other hand, CEM produced slightly more accurate results when supplied with TopHat-aligned reads, particularly when considering all features (gain of r_s = approximately 0.08). Correspondence with CEM's developers revealed that the program requires the TopHat-specific SAM/BAM tag 'XA', which encodes information about the strand of the transcript to which a read aligns, to correctly parse multi-fragment reads. Because this tag was not supplied in our input alignment files, CEM was unable to properly parse alignments that covered splice junctions and therefore produced less accurate estimates when supplied with our alignments.

A subset of the methods (CEM, eXpress, IsoEM, RSEM, and Sailfish) also attempt to correct various biases that occur during sample preparation, such as positional (non-uniform distribution of reads along transcripts), sequencing (depending on the nucleotide composition of the reads), or mapping (sequencing errors and multi-mapping reads) biases (see Methods section for details). While in general we have restricted ourselves to executing each program with the default parameter settings, we wanted to explore whether bias correction had an impact on the abundance estimation (Supplementary Figure A.9). Surprisingly, only the transcript estimates produced by CEM and, to a lesser extent, IsoEM were affected. For CEM, the largest difference was observed when considering expressed transcripts, for which bias correction (default: disabled) had a slight detrimental effect (r_s loss = approximately 0.05). In contrast, the estimates produced by IsoEM seemed to slightly improve upon enabling the bias correction, but only when all transcripts were considered (r_s gain = approximately 0.02). In all other cases, no appreciable differences were observed when executing programs with or without bias correction.

2.3.6 *Gene/transcript structural features affect the estimates of individual methods*

Next, we aimed to assess the impact of gene structural features on the accuracy of expression estimates. Specifically, we sorted transcripts according to their length, proportion of guanines and cytosines nucleotides ('GC-content'), and the number of exons of which they are composed. Likewise, we sorted genes by the number of annotated transcript isoforms. Reasoning that the influence of gene structural features on estimation accuracy is likely to be small compared to that of expression level differences, we concentrated on transcripts with mid-range expression, where differences should be most clearly apparent. For each of the structural features, we then defined non-overlapping bins containing comparable numbers of transcripts or genes. Supplementary Figure A.10 shows the expression level dis-

tributions across the different bins for each of the gene structural features. For each bin we then calculated Spearman correlation coefficients between the 'ground truth' expression and the estimates produced by each of the surveyed methods when supplied with the 30 million read synthetic dataset (Figure 2.3). While none of the analyzed features had a strong and consistent effect on estimation accuracy, we have observed some general trends, as well as method-specific exceptions. The shortest transcripts are quantified with the least accuracy by all methods but Scripture (Figure 2.3 A). This effect cannot be readily explained by differences in expression level distributions across bins, since the smallest transcripts exhibit, in fact, the highest median expression (Supplementary Figure A.10 A). Moreover, the accuracy of isoform-level estimates steadily increases with transcript length for five of the surveyed methods, with eight methods reporting the most accurate estimates for the longest transcripts. Nevertheless, differences in the correlation coefficients are moderate, in the range of approximately 0.04 (BitSeq) to approximately 0.14 (Cufflinks). Similarly, high GC content appears to have a slight, unfavorable influence on the accuracy of isoform abundance estimates, with all but CEM and Cufflinks producing the least and the most accurate estimates for transcripts with high, and low GC content, respectively, and with the differences in the range of approximately 0.02 (BitSeq) and approximately 0.13 (Scripture) (Figure 2.3 B). An intriguing phenomenon becomes apparent when analyzing transcripts according to the number of exons that they contain (Figure 2.3 C): single-exon transcripts are quantified with the least accuracy by all but two methods (Scripture and eXpress). The differences in accuracy relative to bin with the second-lowest accurately are generally small (in the range of approximately -0.01 for BitSeq to approximately -0.05 for CEM) and thus the effect may, at least in part, be explained by the previously described influence of transcript length. However, for Cufflinks this difference is very high (approximately -0.64). Indeed, Cufflinks fails to produce non-zero estimates for the vast majority of single-exon transcripts (Supplementary Figure A.11 A), but not for transcripts containing at least two exons (Supplementary Figure: A.11 B, C). This is not due to an incompatibility between Cufflinks and our read processing/alignment procedure, because applying Cufflinks to TopHat2-generated alignments recapitulates the effect (Supplementary Figure A.11 D, E, F). Interestingly, Scripture exhibits the opposite effect, producing the most accurate estimates for single-exon transcripts (difference to next-best bin approximately 0.11). When excluding single-exon transcripts and apart from Scripture, the influence of exon number is marginal, with differences in accuracy across bins in the range of approximately 0.01 (BitSeq) to approximately 0.05 (rSeq).

Similar to single-exon transcripts, genes with a single transcript isoform that generate just one transcript species are least accurately quantified by most methods except Scripture (Figure 2.3 D). This is to a large extent a consequence of the fact that single-isoform genes are in fact those giving rise to single-exon transcripts (621 of 1'322 genes, that is, approximately 47.0%). Additionally, genes that have only a

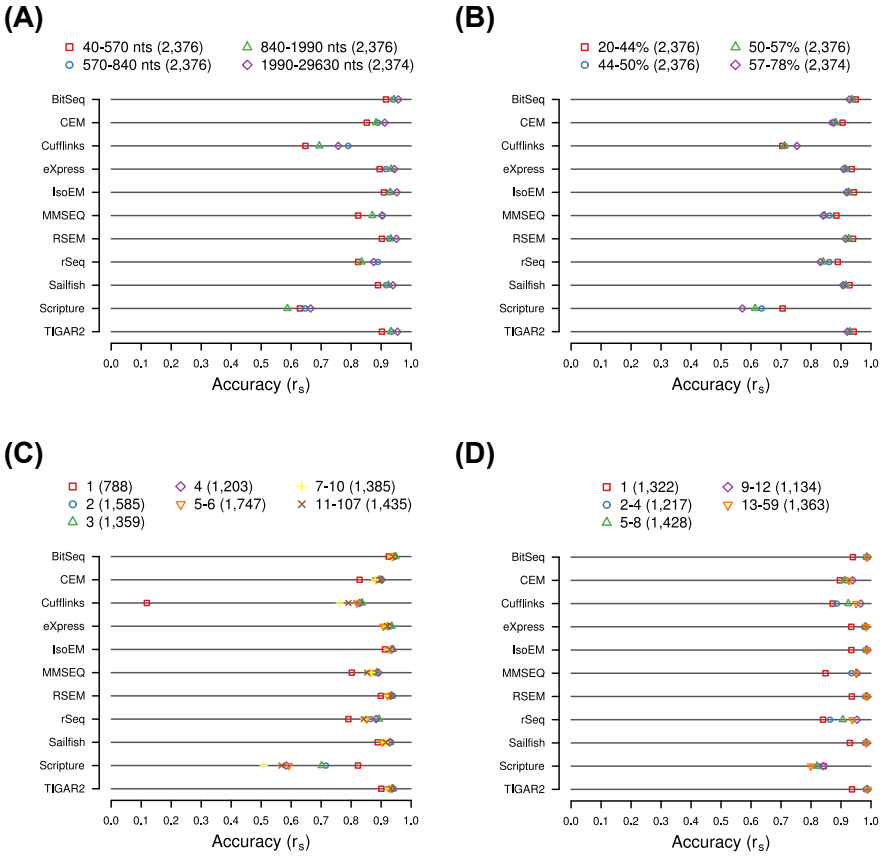


Figure 2.3: Impact of gene structural features on expression estimates. All transcripts or genes expressed at medium levels (0 < log2 TPM <5.5) were distributed across bins according to transcript length **(A)**, GC content **(B)**, the number exons per transcript **(C)**, and the number of transcripts per gene **(D)**. Ranges of the corresponding values covered by each bin are indicated in the legends above each chart. In all cases, expression levels were estimated with each of the indicated methods based on in silico-generated sequencing data (read depth = 30 million). The accuracy of estimates was measured in terms of how well they correlate with true expression levels, expressed as the Spearman correlation coefficient rs, and is indicated for each bin and method

small number of associated transcripts also have low expression levels (Supplementary Figure A.10 D). Otherwise, the complexity of the locus appears to have little impact on the accuracy of isoform abundance estimation: maximum differences in accuracy between bins are in the range of approximately <0.01 (Sailfish) to approximately 0.09 (rSeq), with seven methods exhibiting differences below 0.01. Taken together, our results indicate that, apart from a few method-specific exceptions, the influence of gene structural features on the accuracy of estimates is small. BitSeq, CEM, eXpress, IsoEM, RSEM, Sailfish, and TIGAR2 produce the most robust estimates across the assessed features, with the standard deviations of accuracies across the bins analyzed for each feature being around or below 0.025 (Supplementary Figure A.12). As an additional quantification of the impact of various structural features, the P values of the Kolmogorov-Smirnov's goodness of fit tests carried out for the log-ratio of estimated and expected levels for genes/transcripts in specific bins compared to the entire set of genes/transcripts with moderate expression level (0 < log2 TPM <5.5 and 1.1 < log2 TPM <6.2 for transcripts and genes, respectively; compare categories in Figure 2.2 C, D).

2.3.7 *Isoform- and gene-level estimates are consistent across biological replicates*

A basic test for any inference method is whether they produce similar results when supplied with similar data. For isoform quantification, reproducibility was generally tested on data that was generated synthetically. To investigate this aspect, here we have also prepared RNA-seq libraries from two batches of cells of two cellular systems, the murine fibroblast cell line NIH/3T3 and the human T cell line Jurkat. We then supplied the tools for inferring transcript isoform abundances with the resulting short reads (Sailfish) or alignments (all other tools). The replicate agreement, defined as the Spearman correlation coefficient rs between the estimated abundances of (groups of) transcripts in the two human or mouse replicates, was generally high. At the gene level, rs ranged from approximately 0.82 for both human (Cufflinks) and mouse (MMSEQ) to approximately 0.91 (human; BitSeq) and 0.90 (mouse; Sailfish). In contrast, at the transcript level, the agreement was much lower and varied considerably between tools, in the range of approximately 0.62 (TIGAR2) and 0.60 (MMSEQ) to approximately 0.95 and 0.91 (both Scripture) for human and mouse (Figure 2.4 A and Supplementary Figure A.13 A, respectively). However, only the estimates produced by Scripture and BitSeq showed agreements substantially above rs = 0.7. Most methods produce estimates that are indicative of stronger fluctuations on the transcript compared to the gene level (mean difference in replicate agreement approximately -0.14 and -0.15, for human and mouse), likely because a large proportion of isoforms are expressed at low levels or not at all. In a few cases, differences between replicate agreement on the gene and transcript level exceed 0.2 in at least one species (MMSEQ,

RSEM, rSeq, Sailfish, TIGAR2). On the other side of the spectrum, Scripture exhibits a slightly higher agreement between its transcript than its gene level estimates across both organisms (differences of approximately 0.09 and 0.06 for human and mouse, respectively). These behaviors likely reflect differences in the models underlying different methods, particularly with regard to how they treat low abundance transcripts and how readily they assign reads to the minor and major isoforms of a given gene.

2.3.8 *3′ end sequencing provides independent estimates of isoform abundance*

While the tools for inferring isoform abundance have been quite extensively tested on simulated data, obtaining independent and comprehensive experimental reference data is not trivial. Quantitative PCR (qPCR) is the experimental method of choice for the quantification of transcript abundance. However, despite recent technological advances allowing qPCR experiments on a large-scale level, these methods are still cost- and resource-intensive. We therefore applied our A-seq-2 protocol [25] to prepare 3′ end sequencing libraries from the same RNA preparations that were used for RNA-seq and sought to use 3′ end sequencing-based abundance estimates as an independent experimental reference dataset for assessing the accuracy of expression estimates produced by the benchmarked methods.

To assess the quality of these data we first quantified and compared the usage of annotated 3′ end processing sites that overlap the ends of GENCODE-annotated transcripts (see Methods) between biological replicates. We carried out this analysis both at the level of individual 3′ end processing sites as well as at the gene level. For the latter, we aggregated the abundance estimates of all 3′ end processing sites associated with individual genes. Figure 2.4 B (human) and Supplementary Figure A.13 B (mouse) depict the Spearman correlation coefficients between 3′ end processing site abundances across biological replicates, whereas Figure 2.4 C (human) and Supplementary Figure A.13 C (mouse) show the same on the gene-level. In all cases, the agreement was very high (rs >0.97), suggesting that gene expression and 3′ end processing site usage are highly similar in the replicates that we obtained from both human and mouse cells.

Because in constructing the catalog of 3′ end processing sites from published data we applied stringent validation criteria, the set of 'known' sites is probably biased towards those that are used in relatively abundant transcripts. We therefore wondered whether the agreement between biological replicates is higher when one focuses only on the GENCODE transcripts that end in a 'known', annotated 3′ end processing site and that are likely to be polyadenylated. This was the case for 46′801 human and 26′821 mouse transcripts (corresponding to 25′393 and 17′183 3′ end processing sites, respectively; see Methods section). We selected these transcripts from the output of each method and computed again the correlation between the esti-

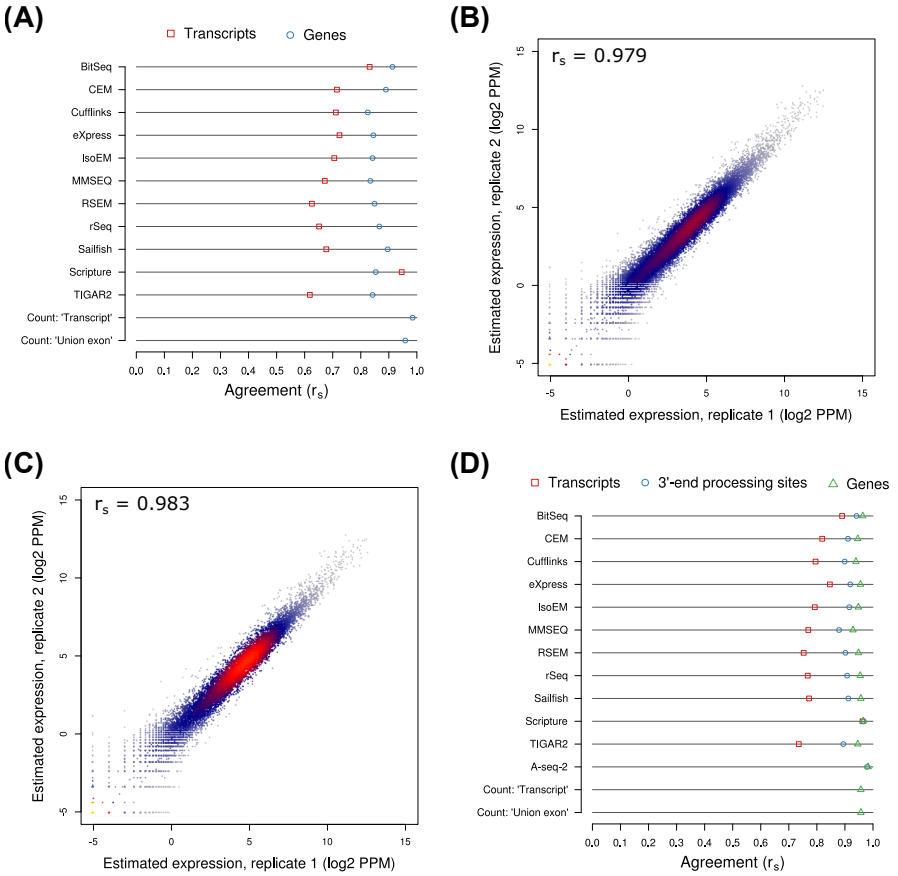


Figure 2.4: Agreement between expression estimates for replicates of Jurkat cells. (A) Transcript isoform and gene expression levels were estimated with each of the indicated methods from two biological replicates of human Jurkat cell RNA-seq data. The agreement between expression estimates of the two replicates are indicated as Spearman correlation coefficients rs, both at the level of transcripts and genes. **(B)** A-seq-2-based 3′ end processing site expression level estimates for the two replicates are plotted against each other. The Spearman correlation coefficient rs is indicated. **(C)** As in (B), but gene level estimates are compared. **(D)** As in (A), but with the addition of 3′ end processing site abundances. For computing expression estimates for either feature type (transcript, 3′ end processing site, and gene), only those transcripts are considered that end in annotated 3′ end processing sites (see main text and Methods for details)

mated levels of transcripts, 3′ end processing sites, and genes (the latter two by aggregation; see Methods section) in the two replicates. Figure 2.4 D and Supplementary Figure A.13 D show the results for the human and the mouse datasets, respectively. As expected, the correlation coefficients computed based on transcripts with annotated 3′ end processing sites were, without exception, higher than those computed based on all GENCODE-annotated transcripts (Figure 2.4 A and Supplementary Figure A.13 A). On the transcript level, Spearman correlation coefficients ranged from approximately 0.74 (TIGAR2) and 0.76 (MMSEQ) to approximately 0.96 and 0.94 (Scripture) for human and mouse, respectively. For 3′ end processing sites and genes, Spearman correlation coefficients of at least 0.88 were reached by all methods for the human and mouse datasets, respectively. The gene expression level estimates provided by the count-based methods also exhibited high agreement (>0.9 for both organisms).

Finally, we further filtered the set of considered transcripts by excluding those whose 3′ ends were not captured in our A-seq-2 dataset. However, in contrast to synthetic data, where the omission of absent transcripts led to a strong increase in estimation accuracy, this did not lead to a further improvement of the correlation between replicate samples (Supplementary Figure A.14 A and B for human and mouse data, respectively). The reasons for this behavior are at the moment unclear. Nevertheless, this analysis indicates that estimates of isoform expression are more reproducible when annotated, and probably more highly expressed poly(A) sites are considered.

Having established that the RNA-seq data lead to highly reproducible estimates of isoform expression, we asked whether the computationally estimated expression levels within individual replicates agree with those that were measured experimentally with the A-seq-2 method. As before, we have aggregated the isoform abundance estimates for each 3′ end processing site and these, in turn, for each gene. Moreover, by selecting 3′ end processing sites that overlapped the end of exactly one transcript, we were able to assess estimation accuracy on the level of individual transcripts. As shown in Figure 2.5 A (human) and 2.5 B (mouse), the expression estimates produced by the surveyed methods are in strong agreement with those based on A-seq-2 across all samples from both human and mouse, with the Spearman correlations approaching those obtained on synthetic data. Agreement between transcript estimates ranges between approximately 0.67 (Cufflinks) and 0.81 (BitSeq) for the human, and approximately 0.71 (Cufflinks) and 0.84 (BitSeq) for the mouse data. When considering 3′ end processing sites that overlapped with the ends of multiple transcripts, correlations further improve, with Spearman correlation coefficients for human and mouse data now in the range of approximately 0.77 (Cufflinks) to 0.86 (BitSeq), and approximately 0.85 (BitSeq) to 0.74 (Cufflinks) respectively. For reference, the corresponding scatter plots for the first replicates of each dataset are presented in Supplementary Figure A.15 (human) and Supplementary Figure A.16 (mouse). Finally, aggregation of 3′ end processing site estimates per gene led to a further increase in agreement by ap-

proximately 0.04 to approximately 0.08 in both organisms. Assuming the A-seq-2-based estimates of expression as 'ground truth', Scripture (rs = approximately 0.92) and RSEM (rs = approximately 0.88) delivered the most accurate estimates at the gene level for human and mouse data, respectively. Importantly, we found that even when estimating gene-level abundance from biological data, isoform-aware methods yield more accurate results than the broadly used count-based methods. Across all methods, level of coarse-graining, and organisms, the second replicate yields estimates that are slightly more accurate, likely reflecting a batch effect pertaining to the preparation of RNA-seq and A-seq-2 sequencing libraries. On all levels, differences in accuracy between most methods are rather small, similar to what we observed on synthetic data. Also similarly, enabling or disabling bias correction in those methods that provide such an option also did not substantially alter the accuracy of estimates on experimental datasets (Supplementary Figure A.17) and in the case of CEM, we have observed a consistent detrimental effect of bias correction across transcripts, 3' end sides, and genes, and in both organisms.

As a practical guideline for those researchers studying non-coding genes, we also wondered how accurately the surveyed methods can quantify the expression of different classes of genes. Therefore, we computed the agreements of expression estimates with those inferred with A-seq-2 on genes annotated as 'protein coding', 'lincRNA' (long intergenic non-coding RNAs), and 'antisense' in both human and mouse. For human, the 12'513 protein coding genes amenable to quantification by A-seq-2 are considerably more accurately quantified than lincRNA (739) and antisense genes (739), with Spearman correlation coefficients reaching values of up to approximately >0.85, 0.8, and 0.7, respectively, for the different gene classes (Figure 2.5 C). The absolute difference in estimate accuracy across these classes is particularly striking for Cufflinks, where the Spearman correlation coefficients are reduced by almost 0.4 when trying to estimate lincRNA or antisense RNAs rather than protein coding genes. This may reflect the issue that Cufflinks seem to have with quantification of single-exon transcripts. Given the differences in median A-seq-2-based expression levels across each gene class (log2 PPM = approximately 3.73, -0.63, and -0.49 for protein coding, lincRNA, and antisense genes and considering both replicates), it is likely that the observed differences in estimation accuracy are, at least in part, a function of the true expression levels of these genes. Although the general trend is the same across the mouse samples, the differences in estimation accuracies between the different gene types are not as pronounced as in human, and for some methods the quantification of lincRNA genes is actually more (rSeq, Scripture) or approximately equally accurate (BitSeq, TIGAR2) as that of protein coding genes (Figure 2.5 D). This may reflect the true abundance of these genes because A-seq-2 estimates of the median expression of lincRNA and antisense gene classes were somewhat higher for mouse (log2 PPM = approximately 0.00 and 0.10, respectively) while those for protein coding genes were about the same (median log2 PPM = approximately 3.67). Taken together, the

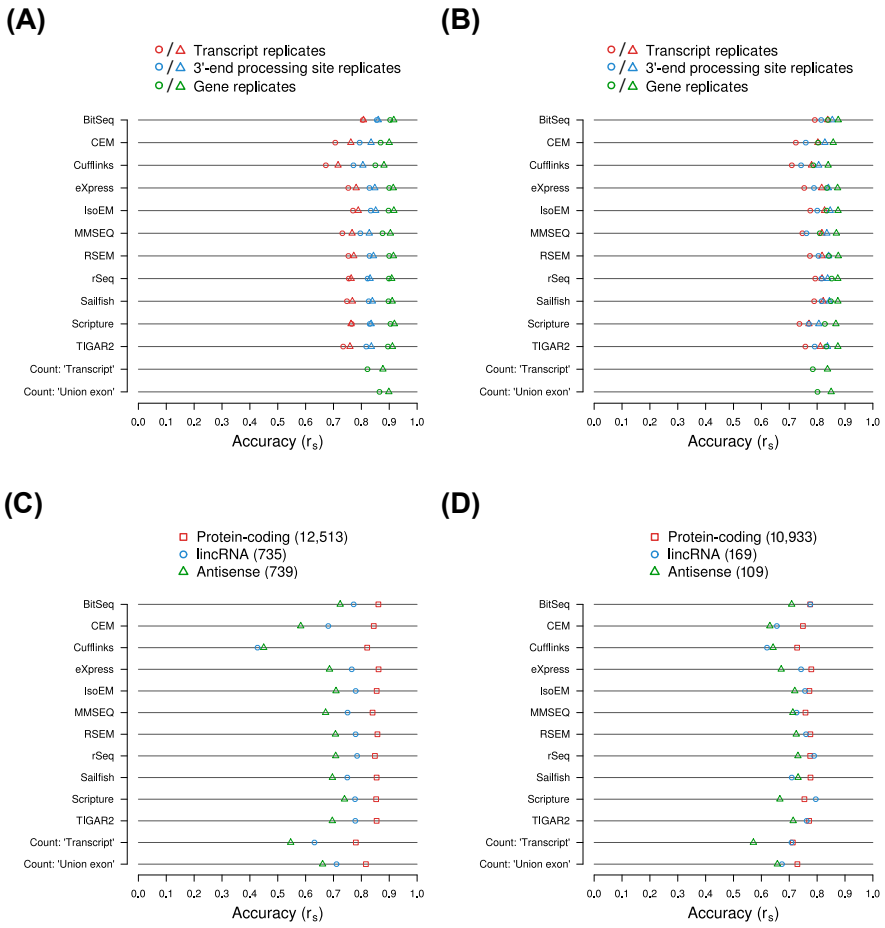


Figure 2.5: Agreement between the expression level estimated computationally from RNA-seq data and those measured with an independent experimental method. (A) and **(B)** Abundances of 3′ end processing sites in two independent samples (circles: replicate 1, triangles: replicate 2) of human Jurkat (A) or murine NIH/3T3 cells (B) were quantified with A-seq-2. Based on RNA-seq data obtained the same cell cultures, the abundances of transcripts ending at these processing sites were estimated with each of the indicated methods and aggregated per processing site. 3′ end processing site estimates were further aggregated per gene. The agreement between A-seq-2- and RNA-seq-based expression estimates was computed as Spearman correlation coefficients (rs) for 3′ end processing sites, genes, and transcripts (when processing sites were associated with exactly one transcript). Refer to the main text and the Methods section for further details. **(C)** and **(D)** Similar to (A) and (B), but only gene expression level estimates were considered and Spearman correlation coefficients were computed independently for different classes of gene biotypes, both for the human (C) and mouse (D) data. Plotted data represent means of the Spearman correlation coefficients calculated for each of two replicates

estimates of isoform expression based on biological data and evaluated against expression measurements obtained with an independent experimental method validate and recapitulate the most important conclusions derived from the synthetic data: many of the surveyed methods are able to estimate isoform abundances with good accuracy, particularly when true expression levels are high. Furthermore, employing any of these tools improves the accuracy of gene expression level estimates relative to widely used count-based methods.

2.4 DISCUSSION

Accurate quantification of gene expression is essential for the understanding of gene regulatory processes in health and disease. Due to its large dynamic range, high reproducibility, and the ability to detect previously unknown transcripts, RNA sequencing has become the method of choice for global expression profiling. However, despite the digital nature of the resulting data, technical limitations (limited read length and non-uniform transcript coverage) render their analysis challenging, especially when large and complex genomes of higher eukaryotes, with frequent repeats and overlapping gene structures, are involved. Accurate computational methods for RNA-seq data analysis therefore remain in high demand. This is reflected in the large number of computational methods for estimating transcript isoform abundance that were developed over the course of the last 6 years. Naturally, the question arises which method should best be used in a particular context. Here we have tried to address this question in depth, using not only synthetic data, as is typically done when the computational methods are developed, but also using estimates that were obtained with an independent experimental method for a specific type of isoforms, namely those that arise from alternative polyadenylation. This is because methods for global quantification of 3′ end site usage distinct from RNA-seq are available [16] [151] [152] [153] and have been used quite extensively to analyze changes of 3′ UTR isoforms across conditions. A drawback of these methods is that they cannot distinguish between transcripts that are processed at the same poly(A) site. However, although most mammalian genes have multiple poly(A) sites, currently, over 60% of the poly(A) sites whose expression we have been able to measure with A-seq-2 have only one associated transcript annotated in the human or mouse GENCODE datasets. Thus, we believe that A-seq (or another method for quantifying the usage of 3′ end processing sites) can offer a good alternative to qPCR as a comprehensive approach to transcript isoform quantification. Nevertheless, as the 3′ end sequencing protocols are relatively new, it is likely that the computational analysis of these data can be further improved.

Expecting that most users would – at least initially – run the methods 'out-of-the-box', we sought to apply the surveyed methods with default settings, and departed from this general rule only to test the influence of specific options that the developers of the methods pro-

posed. Although we found that neither the use of recommended short read aligners nor the activation of bias correction generally improved estimation accuracy, it is likely that the developers of the individual methods or experienced users would be able to improve the performance of individual tools in specific settings. During the course of this study we discovered a number of assumptions that the programs tacitly made and that affected the interpretation of the results. Therefore, a specific recommendation that we can make to developers is to ensure that sufficiently detailed information on input requirements, potential pitfalls and the implication of specific options (ideally including usage examples) is provided.

Encouragingly, we found that most of the methods that are currently used to estimate transcript isoform abundance produce quite comparable and accurate results, both on synthetic and experimental data. As a general trend, methods such as Scripture and Cufflinks, whose main objective is to assemble/reconstruct transcript isoforms but that have also been co-opted for estimating isoform abundance, perform poorer than methods specifically designed for the latter purpose. However, such methods could be part of the initial assembly of a comprehensive set of transcripts whose expression can be subsequently quantified with a different approach [66]. Cufflinks is part of the popular 'Tuxedo Suite' pipeline (Bowtie-TopHat-Cufflinks) and for the purpose of inferring isoform abundances from RNA-seq data is probably superseded by the eXpress method developed by the same group [139]. Importantly, the gene level expression estimates obtained by cumulating the abundances of transcript isoforms inferred with almost any of the surveyed methods are more accurate than those produced by 'count-based' methods that are widely used in the analysis of gene expression. This is likely because count-based methods either disregard or mis-assign reads whose origin (genomic locus or isoform) cannot be unambiguously determined. We therefore strongly advise to use methods for transcript isoform quantification (such as those benchmarked here) even when only quantification at the gene level is desired.

Next to a general assessment of the accuracy of expression estimates produced by the tools, we also studied the impact of several transcript properties on the accuracy of expression estimation. We found that parameters that directly influence the coverage of a transcript or gene by reads, such as sequencing depth and true expression level, have a positive influence on estimation accuracy, as has been observed before [154]. On synthetic data, disregarding features that are not expressed led to a strong increase in the accuracy of expression estimates, particularly on the level of isoforms. Thus, as may be expected, estimates of low-abundance isoform expression are not very reliable. How isoforms that are expressed at very low levels (or not at all) are treated in practice, varies between methods. Most methods report (or imply) cases of 'zero' expression and some allow the user to specify a minimum level of expression for reported transcripts. On the other hand, BitSeq and MMSEQ do not enforce such a threshold and instead attempt to assign non-zero priors even to transcripts that

are not supported by any read, based on factors such as the library size and transcript length. These solutions represent lower and upper bounds on the expression of low-abundance transcripts (in contrast to higher-abundance transcripts, for which precise estimates of expression are sought). In typical RNA-seq experiments, where many transcripts are expected to be expressed, how precisely absent transcripts are treated may not be essential. However, in the case of, for example, single cell sequencing, the proportion of annotated transcripts that are not detected can be quite large and one should be aware that the meaning of the expression values that the programs report are not entirely the same for expressed and not expressed transcripts. Next to coverage-related factors, we found that the length and GC content of transcripts as well as the complexity of the gene locus (exons per transcript and transcripts per gene) have a small impact on the accuracy of inferred expression levels, which is probably of more interest to the developers rather than to the average user.

To ensure the widest applicability of our findings, we have based our study on single-end, short read (50 nt) data. Illumina's paired-end sequencing technology, which has been employed in previous comparisons of isoform abundance estimation methods [132] [154], provides additional information that may be used by many of the evaluated methods to improve the assignment of read fragments to the correct isoform and thereby the accuracy of abundance estimates. As has been previously demonstrated [154] [142], [143], increasing the read length should also enhance the accuracy of abundance estimates, because it leads to a reduction in the fraction of reads that cannot be unambiguously assigned to the correct isoform. Indeed, increasing the read length is a current trend in the field of next generation sequencing. For example, Pacific Biosystems technology now allows full-length transcript sequencing [129], although at limited throughput.

While most methods produce comparable and fairly accurate estimates of transcript isoform abundance, they differ more strongly in their computing needs. In some cases, speed comes at the cost of increased memory requirements, which is evident for example with IsoEM, which is extremely fast, but uses tens of GB of memory. Nonetheless, with the increase in the number and size of the datasets that one typically analyzes, speed and scalability of processing become very important considerations for the utility of a program. The recently developed Sailfish is of particular interest in this regard because its running times scale well within the tested range of sequencing depths, while the memory footprint remains reasonable. Moreover, its alignment-free k-mer-based approach disposes of the time-consuming step of aligning reads to a reference genome or transcriptome. For typical datasets of approximately 100 million reads, most programs use 1–20 GB of memory and run for 1–2 h. An exception is TIGAR2, which produces highly accurate expression estimates that come at the cost of both high running times and high memory use.

One important aspect that was beyond the scope of the current study is that in many studies, the interest is the identification of transcript isoforms that are differentially expressed between two conditions, rather than the quantification of isoform abundance in a specific condition. The estimates of isoform abundance inferred with the methods that we tested here can in principle be used in subsequent statistical tests for differential expression, but the issue of the underlying model has not been entirely addressed. If sufficient replicates are available, two-sample parametric or non-parametric tests can be used. However, due to the high costs of RNA-seq experiments, the availability of more than a few replicates is very rare. Instead, when the number of replicates is small, accurately accounting for the different sources of variability in the data is important. Differential expression analysis based on RNA-seq data is frequently done with programs such as baySeq [155], DEGSeq [156], DESeq [60], or edgeR [59] (reviewed in [157]). These programs work on (integer) count data and use specific models for the number of reads that are expected from individual 'features' such as exons or genes. Therefore, they are not appropriate for the estimate of transcript abundances that are obtained with the programs that we analyzed here. Fortunately, some of the evaluated programs have additional modules for differential expression analysis. BitSeq has a built-in functionality for differential expression analysis based on the transcript expression levels estimated by the tool. The developers of Cufflinks and eXpress suggest Cuffdiff [158] for gene and transcript differential expression based on their respective outputs. The developers of IsoEM suggest the bootstrapping-based IsoDE [159] for differential expression analysis, but this tool is restricted to comparisons at the gene-level only. MMSEQ's developers suggest MMDIFF [160] which performs model comparisons and takes as input the posterior summaries from the MMSEQ tables. Alternatively, they provide instructions to feed MMSEQ-estimated counts to count-based differential expression analysis tools like DESeq or edgeR `https://github.com/eturro/mmseq/blob/master/doc/countsDE.md`. eXpress and Sailfish developers also suggest to feed the supplied (rounded) 'effective counts', and 'expected number of reads', respectively, into one of the count-based differential analysis tools mentioned above. Finally, RSEM developers suggest EBSeq [161], a Bayesian differential expression analysis method for genes and isoforms across two or more biological conditions. EBSeq is integrated into the RSEM suite `http://deweylab.biostat.wisc.edu/rsem/README.html#de`.

2.5 CONCLUSIONS

In summary, several methods for the inference of transcript isoform abundance can accurately quantify expressed transcripts even from relatively small short-read libraries and should thus be adequate for the analysis of both past and present RNA-seq datasets. Their performance is largely not affected by structural features (number of exons,

transcript length, GC content) of the genes/transcripts, although, as expected, abundant transcripts are quantified more accurately compared to rare transcripts. Importantly, our analysis indicates that the explicit quantification of transcript isoforms leads to more accurate estimates of gene expression levels compared to the 'count-based' methods that are broadly used currently. Given the wealth of tools available, the user can largely base his choice of method on criteria related to usability, available processing and memory capacities, compatibility with pre-existing data processing pipelines, and the desired downstream analyses (see Table 2.2). Especially promising is the most recently proposed approach that relies on k-mer frequencies, bypassing entirely the read-to-genome/transcriptome alignment and thereby enabling analysis of very large collections of samples, such as those that have started to emerge from patient studies. Developers may profit from our study setup, particularly our efforts to provide compatible datasets to tools with quite different requirements as well as our approach at validating estimation accuracies of a particular type of isoform with an independent large-scale experimental method. We propose that methods such as 3 end sequencing and cap analysis of gene expression (CAGE; [162]), which allow quantification of alternative polyadenylation and transcription start sites, respectively, could complement validation schemes based on synthetic data and quantitative PCR in future or ongoing assessments of RNA-seq analysis methods, such as, for example, by the MAQC-III/RNA-C consortium [163].

2.6 METHODS

2.6.1 *Genomes, gene annotations, and transcriptome sequences*

The hg19 (human) and mm10 (mouse) genome assemblies were obtained from UCSC Genome Bioinformatics, University of California, Santa Cruz `https://genome.ucsc.edu/index.html`. Haplotype chromosome versions were discarded. Releases 19 and M2 of the GENCODE gene annotation sets GENCODE [26] were used for the analysis of human and mouse data, respectively. Version numbers were stripped from gene and transcript identifiers. In the human annotation set, all features on the Y chromosome that are present, in identical form, on the X chromosome have gene identifiers of the form 'ENSGRx' (with x being a 10-digit number), and the corresponding features on the X chromosomes have identifiers of the form 'ENSGox'. We discarded the former to avoid essentially duplicate features. Sequences of annotated transcripts ('transcriptomes') were obtained from ENSEMBL (release 74, compatible with GENCODE v19 and vM2) [164]. Genome and transcriptome sequences in FASTA format were indexed with segemehl [52].

Table 2.2: Features and performance summary of the surveyed methods. To facilitate a user's choice of method, we indicate which methods meet various criteria of usability, functionality, and performance, as follows: 'Extensive documentation' - documentation that goes beyond the description of parameters is provided (document, web page, FAQ which allowed us to run a given method confidently and without help from developers); 'Standard file formats' - the method exclusively operates on the indicated file formats for transcript sequences (FASTA), gene/transcript annotations (GFF/GTF or BED12), read sequences (FASTA or FASTQ), and read alignments (SAM/BAM as defined in [45] and produced by most modern aligners); 'Gene-level estimates' - estimates of expression on the gene level are provided in addition to those at transcript level; 'Reconstruction supported' - the method can also reconstruct transcript models based on the provided sequencing/alignment data; 'DE analysis' - the developers make a general recommendation or provide an integrated solution for differential analysis of transcript/isoform expression; 'Efficient multi-threading' - the method efficiently makes use of multiple cores (speedup of at least two-fold in at least three out of five datasets; see Supplementary Figure A.2 A); 'Fast' - processing of 100 million synthetic reads or their corresponding alignments completed in less than 1 h (16 cores and 64 gigabytes provided; see Figure 2.1 B); 'Small memory footprint' - all synthetic datasets could be processed with less than 8 gigabytes of memory (independent of the number of cores used; see Figure 2.1 C, D). Additional details are provided in the main text.

Method	Extensive documentation	Standard file formats	Gene-level estimates	Reconstruction supported	DE analysis	Efficient multi-threading	Fast	Small memory footprint
BitSeq	X	X			X	X		
CEM		X		X			X	X
Cufflinks	X	X	X	X	X	X		
eXpress	X	X			X			X
IsoEM		X	X			X	X	
MMSEQ	X	X	X		X			X
RSEM	X		X		X	X		
rSeq	X		X					X
Sailfish	X	X			X	X	X	X
Scripture	(X) [3]	X		X				
TIGAR2	X	X						

2.6.2 *Generation of synthetic sequencing data*

To generate in silico reads, we have used the Flux Simulator software [148], with the hg19 genome and GENCODE v19 annotation set processed as described above. Because we focused on the quantification of long RNAs, we further removed from the annotation set, all entries whose gene or transcript type attributes matched either 'miRNA', 'misc_RNA', 'rRNA', 'snoRNA', 'snRNA', 'Mt_rRNA', or 'Mt_tRNA'. Taking into account the annotated transcripts introduced above as well as a target number of transcript molecules (we chose 5 million), Flux Simulator randomly assigns expression ranks to transcripts according to Zipf's Law. The software then attempts to model the various steps in a typical RNA-seq library preparation protocol, including fragmentation, reverse transcription, and PCR amplification, to generate reads. We ran Flux Simulator with the options –express, –library, and –sequence. Additional parameters were supplied in a parameter file as outlined in the Flux Simulator manual `http://sammeth.net/confluence/display/SIM/Home`. Flux Simulator does not natively support generation of directional single-end read libraries. To obtain these, we instead generated a pool of 692,414,670 paired-end reads from which we then discarded all antisense mate sequences, as suggested by the Flux Simulator developers. To facilitate downstream processing, the identifiers of the remaining reads were simplified and their sequences capitalized. Identical read sequences were collapsed with the fastx_collapser `http://hannonlab.cshl.edu/fastx_toolkit/index.html`. Finally, poly(A)-tails - introduced in the simulation - were removed with the cutadapt software [38] by specifying a stretch of 50 adenines as the 3' adapter and the non-default options –overlap=1 and –minimum-length=15. This resulted in a set of 298,435,172 poly(A)-free, directional, single-end reads. From this initial set, we randomly selected, progressively, approximately 100 (100,001,950), 30 (30,004,152), 10 (10,000,760), 3 (2,998,971), and 1 (999,436) million reads to analyze the scaling behavior of the programs.

2.6.3 *Preparation of sequencing libraries*

Human Jurkat T lymphocytes (ATCC TIB-152) [165] and NIH/3T3 mouse embryonic fibroblasts (ATCC CRL-1658) [166] were cultured in RPMI medium (Sigma) at 37 Celcius and 5% CO2. Cells were collected at approximately 70% confluency after trypsinization. 3' end libraries were generated by the A-seq-2 protocol, which captures sequences immediately upstream of mRNA 3' end processing sites and poly(A)-tails [167], and directional RNA-seq libraries were prepared according to the Illumina-provided protocol. For both protocols, poly(A)-positive RNA was isolated from the cells with the 'Dynabeads mRNA DIRECT Kit' (Ambion) and fragmented by alkaline hydrolysis to fragment sizes of 150-300 nt. Following reverse transcription and PCR amplification, the libraries were sequenced single-end with a read length of 51 nucleotides on an Illumina HiSeq-2000 platform.

2.6.4 *Pre-processing of human and mouse RNA-seq data*

Potential 3′ adapter and poly(A)-tail fragments were sequentially removed from FASTQ-formatted short reads sequences with two iterations of cutadapt [38], specifying the 3′ adapter sequence and a stretch of 50 adenines, respectively, to the –adapter option. Other non-default options were –overlap=1 and –minimum-length=15. Identical sequences were collapsed with the fastx_collapser `http://hannonlab.cshl.edu/fastx_toolkit/index.html`.

2.6.5 *Alignment of synthetic and experimentally obtained reads to genomes and transcriptomes*

The experimentally obtained sequence sets, as well as the five in silico-generated read subsets (FASTA-formatted), were aligned to the genome and transcriptome of the respective species with segemehl 0.1.7 [52], with default parameters (minimum percentage of matches: 90%) and without using the spliced alignment option. Anti-sense alignments to transcripts were discarded from further analysis. For the surveyed methods that require input alignments in 'genome space', transcriptome alignments were converted to genomic coordinates with custom scripts based on the gene models provided in the GENCODE v19 annotation file. Directly and indirectly obtained genome alignments in SAM format were merged, duplicate alignments resulting from the conversion between transcript and genome coordinates were discarded, and the remaining alignments were filtered such that for each read only the alignments with the smallest edit distance were kept. For methods requiring input alignments in 'transcriptome space', the transcriptome alignments of each reads that had an edit distance larger than the minimum distance obtained in aligning the read to the genome were discarded.

During the course of the study, we have noticed that the transcript isoform quantification methods that we evaluated make certain assumptions about the format of the input alignment files and that in some cases these assumptions only hold for certain short read aligners or for outdated file formats. We therefore implemented additional post-processing steps to ensure that the information required by individual programs is present in the alignment file is the appropriate form. (1) We 'uncollapsed' the reads: across all alignment files, alignments corresponding to collapsed reads were 'cloned', but a randomized QNAME name was assigned to each individual read that was only re-used for additional alignments of the same read. (2) To avoid misinterpretation of tag fields, all custom segemehl tags were removed. (3) Reads aligning to more than one reference locus are reported by segemehl as individual alignment records with identical read names (QNAME field). In accordance with the SAM specifications (`http://samtools.github.io/hts-specs/SAMv1.pdf`), we have further added a linked-list encoding for such reads. Specifically, we have designated the first out of such a group of alignments as the

primary (0x100 bit of the FLAG field unset) and introduced CC and CP tags, pointing, respectively, to the reference sequence name and the starting position of the following alignment. All remaining alignments were designated secondary (0x100 bit set), and CC and CP tags were added to all alignments but the last in the list. Moreover, the HI (0-based 'hit index') tag was added to all alignments of 'multi-mapping' reads. The NH ('number of hits') tag was re-computed for all reads in a given alignment file. (4) segemehl reports a default mapping quality (MAPQ) of 255 for each alignment record. Following the example of TopHat2 [168], we have reset the mapping quality values based on the number of alignments reported for a given read. Specifically, we have assigned mapping qualities of 50 (NH = 1), 3 (NH = 2), 1 (NH = 3 or 4), and 0 (NH = 5 or more). (5) We introduced sequencing quality strings (QUAL field). For in silico-generated reads, which did not have such scores associated, strings of 'I' characters that match the length of the read sequence (SEQ field) were used to denote maximum quality scores (according to the Sanger FASTQ format). In the case of the experimental RNA-seq libraries, we used the quality scores that were provided in the initial FASTQ files that were obtained from the sequencing facility. The data processing was automated with the help of the Anduril [29] data analysis framework. To test the influence of the alignment program, we have also generated alignments of in silico generated reads with Bowtie (version 1.0.0) [169] and TopHat2 (version 2.0.10) [168]. The output of these programs were used without further processing.

2.6.6 *Analysis of 3′ end sequencing data*

The reads obtained with the A-seq-2 protocol for 3′ end sequencing have a particular structure: they are the reverse complement of 3′ end RNA fragments and further have the sequence AAANNNN downstream of the actual 3′ end [167] for details). To recover the mRNA 3′ ends from these sequenced reads, we therefore first trimmed the expected NNNNTTT sequences from the 5′ ends of the reads, removed the 3′ adapter with the removeAdaptor.pl function of the CLIPZ server [28] and kept only sequences longer than 15 nt. We reversed complemented the sequences and mapped them to the corresponding genome and transcriptome with segemehl v0.1.7 [52] and default parameters. Next, we transformed transcriptome alignments to genomic coordinates, merged them with the genome alignments, discarded duplicates and kept for each read only those alignments with the smallest edit distance (see above). Finally, we collapsed the 3′ ends of the aligned short reads and produced a BED file recording the exact genomic positions of 3′ end cleavage together with the aggregated read counts. For reads that mapped to multiple loci in the genome, counts were equally distributed across loci. As we and others observed before, 3′ end formation appears to occur with a certain degree of microheterogeneity, that is, prominent 3′ end sites are usually being flanked by less frequently used 3′ end sites. Be-

cause these latter sites may not reflect functional biological variation, closely spaced 3′ end sites are typically clustered into 3′ end processing regions [16]. Many 3′ end sequencing protocols capture sequences that result from priming at internal adenosine stretches rather than poly(A)-tails at the step of cDNA synthesis. To exclude a protocol-specific bias in 3′ end quantification, we only analyzed 3; end processing sites that are supported by at least two independent 3′ end sequencing protocols. These are annotated in our in-house polyAsite database [94]. For each 3′ end processing region, we determined the number of overlapping A-seq-2-inferred 3′ end reads, which we used as a measure of the expression of the corresponding 3′ end processing region. In total, we quantified the expression of 90,128 and 61,457 3′ end processing regions in human and mouse, respectively.

2.6.7 *Estimation of transcript isoform abundance*

With the exception of Sailfish (see below), all of the programs compared in this study use alignments of reads to either the transcriptome or the genome. We used the samtools [45] suite to sort or interconvert the SAM/BAM alignment files obtained from mapping reads to genomes/transcriptomes. To detect and quantify 'multi-mapping' reads, several methods require that the alignment files are ordered such that the alignments of a given read occur one after the other. Additionally, some methods further require that reads that are similar in sequence (and their associated alignments) are randomly distributed in the input file. This is of clear relevance for eXpress, which processes alignments 'on-line' and trains its parameters from the data. In such cases, 'non-random' occurrence of the read alignments may lead to biased parameters and output. Typically, both of these conditions (reads occur in random order while all alignments of a given read are grouped together) are met when alignments are sorted by the names of the reads, which is recommended in the documentation of these methods. But if the pre-processing pipeline includes sorting and renaming steps (for example, collapsing and uncollapsing of reads with identical sequences), sorting the alignment file by read names may lead to a situation in which neither condition is fulfilled. Unfortunately, the precise assumptions about the order in which read alignments should appear in the input file are not typically mentioned in detail in the documentation of the programs. We thus recommend that users ensure that the order in which reads appear in the alignment file that is used as input to an isoform quantification method is 'randomized' whenever the quantification method recommends sorting alignments by read name.

Scripture and CEM require annotation files in a BED-based format which supports multiple fragments (that is, exons) per entry and is known as BED12 or BED12+3. These were generated from the GENCODE-provided GTF annotation files with the help of the R/Bioconductor package rtracklayer [170]. Because some methods required the mean and standard deviation of the fragment/read length

distribution, we calculated these from the alignment files with a custom script. In the following, the steps taken to execute each surveyed program are outlined.

BitSeq [135] [136] uses as input transcript sequences in FASTA format and alignments of reads to the transcriptome in SAM or BAM format, sorted by read name (randomized). We have used the command-line version of BitSeq (version 0.7.5), but an R/Bioconductor version is also available.

The first step in BitSeq is to parse the alignment file to calculate probabilities of individual reads originating from individual transcripts:

```
parseAlignment <alignments_transcriptome> \
--trSeqFile <sequences.fa> \
--outFile <out_prefix.prob> \
--trInfoFile <out_prefix.trx> \
--uniform
```

Then the mean transcript expression is estimated with a Variational Bayes inference algorithm:

```
estimateVBExpression <out_prefix.prob> \
--outPrefix <out_prefix> \
--outType RPKM \
--trInfoFile <out_prefix.trx> \
--samples 1000 \
--seed 1
```

By default, when no read alignments are assigned to a given transcript, BitSeq sets the expression estimate of the transcript to a 'prior' that depends on the effective transcript length and the sequencing depth. When indicated and in communication with the developers, we have identified these cases by finding transcripts for which the expected read count (alpha parameter of the Dirichlet distribution) equals exactly 1 and replaced their RPKM estimates with zeros.

CEM [137] takes as input a BED12 file of transcripts and a SAM or BAM file of genomic alignments, sorted by genomic coordinates. We ran CEM (processsam version 2.5.2) as follows:

```
python runcem.py \
--annotation <annotations.bed12> \
--forceref \
--prefix <out_prefix> \
<alignments_genome.bam>
```

Where indicated, we have set the –usebias option to evaluate CEM's built-in bias correction functionality, which attempts to correct for positional, sequencing, and mappability biases.

Cufflinks [138] takes as input an annotation file in GTF format and a SAM or BAM file of read alignments to the genome, sorted by genomic coordinates. We ran Cufflinks version 2.1.1. with the following command:

```
cufflinks \
--GTF <annotations.gtf> \
--library-type fr-secondstrand \
--frag-len-mean <fragment_length_mean> \
--frag-len-std-dev <fragment_length_sd> \
```

```
--multi-read-correct \
--output-dir <out_dir> \
<alignments_genome.bam>
```

Only expression estimates with 'fpkm_status' 'OK' were considered. All other estimates were set to zero.

eXpress [139] takes as input a FASTA file of transcript sequences and a SAM or BAM file of transcriptome alignments, sorted by read name (randomized). We ran eXpress version 1.5.1. with the following command:

```
express \
--no-update-check \
--f-stranded \
--frag-len-mean <fragment_length_mean> \
--frag-len-stddev <fragment_length_sd> \
--output-dir <out_dir> \
<sequences.fa> \
<alignments_transcriptome.bam>
```

As eXpress is correcting for biases introduced during library preparation (specifically, fragmentation and priming) by default, we have set the –no-bias-correct option when evaluating the performance of methods without bias correction.

IsoEM [140] takes as input a GTF file with transcript annotations and a SAM file of genomic alignments, sorted by read name (randomized). We obtained instructions for running IsoEM from (https://dna.engr.uconn.edu/software/IsoEM/README.TXT) and ran the program (version 1.1.1) as follows:

```
isoem \
--GTF <annotations.gtf> \
--fragment-mean <fragment_length_mean> \
--fragment-std-dev <fragment_length_standard_deviation> \
--directional \
-o <out_file> \
<alignments_genome.sam> \
```

IsoEM also attempts to correct for fragment sampling biases resulting from random hexamer priming during reverse transcription and to evaluate this functionality, we have generated isoform abundance estimates with the -b option.

MMSEQ [141] (version 1.0.8) takes as input a file with transcript sequences in FASTA format as well as a BAM file with read alignments to the transcriptome, sorted by read name (randomized). We ran MMSEQ based on the provided instructions (https://github.com/eturro/mmseq). In particular, we first mapped reads to transcripts:

```
bam2hits <sequences.fa> \
<alignments_transcriptome.bam> > <hits>
```

and then obtained expression level estimates via:

```
mmseq <hits> <out_prefix>
```

Note that unlike all other methods, MMSEQ does not report RPKM values, but rather the means μ of the posterior isoform expression distributions. As these are reported as log (base e) values, we first

exponentiated them for our analyses. Similar to BitSeq, MMSEQ defaults to assigning 'prior' expression estimates to those transcripts for which no read/alignment evidence can be found. Where indicated, and in communication with the developers, we have identified such cases by substituting in MMSEQ's output the log μ estimates for all transcripts or genes with a 'unique_hits' count of 0 with 'NA'.

RSEM [142] [143] (version 1.2.18) works on alignments of reads to transcripts (sorted by read name/randomized in SAM or BAM format). Based on GENCODE annotations, we first generated a tab-delimited lookup table between ENSEMBL gene (first field) and transcript IDs (second field). For each organism (human or mouse), we then generated RSEM-specific indices of the corresponding ENSEMBL transcript sequences (FASTA) with the following command:

```
rsem-prepare-reference \
--no-polyA \
--transcript-to-gene-map <gene_id_transcript_id_table> \
<sequences.fa> \
<index_prefix>
```

RSEM requires read alignments to the transcriptome. However, because the tool cannot process alignments that contain insertions or deletions (indels), we purged the alignment file of any entries that contained disallowed characters in their CIGAR string fields (D, H, I, N, P, S). After recalculating read length distributions across the resulting alignment files, we estimated maximum likelihood expression levels as follows:

```
rsem-calculate-expression\
--sam \
--strand-specific \
--no-qualities \
--seed-length 15 \
--fragment-length-mean <fragment_length_mean> \
--fragment-length-sd <fragment_length_sd> \
<alignments_transcriptome.sam> \
<index_prefix> \
<out_dir>
```

To evaluate RSEM's built-in bias correction functionality, which attempts to correct protocol-specific 5′ or 3′ positional biases, we have set the –estimate-rspd (read start position distribution) option where indicated.

rSeq [144] takes as input a FASTA file of transcript sequences and a SAM file with read-to-transcript alignments, sorted by transcript names and coordinates. Because the header for each transcript in the sequence file is expected to be of the form 'gene_id ' 'transcript_id', we used custom scripts to construct these identifiers and substitute the reference sequences in the sequence dictionary and alignment entries of the transcriptome alignment file accordingly. We then obtained rSeq-based (version 0.2.0) isoform expression levels with the following command:

```
rseq expression_analysis \
<sequences.fa> \
```

```
<alignments_transcriptome.sam>
```

Sailfish [145] (version 0.6.3) takes as input transcript sequences in FASTA format and sequenced reads in FASTQ (or FASTA) format. Sailfish does not required reads to be ordered in a specific manner. The first step in running Sailfish is to index the transcriptome sequences:

```
sailfish index \
-t <sequences.fa> \
-o <index> \
-k 20
```

and then the isoform abundance estimates are obtained with the following command:

```
sailfish quant \
-i <index> \
-l T=SE:S=S \
-r <reads> \
-o <output_prefix>
```

Sailfish considers transcript length, GC content, and dinucleotide frequencies as possible sources of bias and uses a regression model to correct for them. By default, Sailfish reports its output both with and without these 'bias correction' settings. Unless otherwise noted, we have used the estimates without bias correction.

Scripture [106] (archive ScriptureScorer.jar provided by the developers on 6 March 2014) is a tool that was designed for reconstructing and estimating the relative likelihoods of different isoforms. Scripture takes as input a file of transcript annotations (in BED12 format) and a SAM or BAM file with read-to-genome alignments, indexed and sorted by coordinates. We ran Scripture based on instructions provided to us by its developers as follows:

```
java -Xmx<XX>g -jar ScriptureScorer.jar \
-annotations <annotations.bed> \
-alignments <alignments_genome.bam> \
-strand <first> \
-singleEnd \
-minMappingQuality 5 \
-out <out_file>
```

TIGAR2 [146] [147] (update from 6 March 2014) takes as input a FASTA file of transcript sequences and a SAM or BAM genome alignments file, sorted by read name (randomized). We used the following command to run TIGAR2:

```
java -Xms<XX>g -Xmx<XX>g -jar Tigar2.jar \
<sequences.fa> \
<alignments_genome.bam> \
--alpha_zero 0.1 \
<out_file>
```

2.6.8 *Normalization and stratification of expression 'ground truths' and estimates*

In order to assess the accuracy of expression level estimates, we first converted the 'ground truth' transcript abundances provided in the Flux Simulator output for the simulated data and the by the A-seq-2 data (processed as described above) for the human and mouse samples to a standard library size of 1 million reads. We refer to these measures as transcripts per million transcripts (TPM) and processing regions per million processing regions (PPM), respectively. Since the benchmarked methods already supplied estimates in normalized expression units, no further processing of these values was required. In particular, we have used the reads/fragments per kilobase of exon model per million mapped reads (RPKM/FPKM) units wherever available, thus accounting not only for differences in library sizes but also for differences in transcript lengths. The latter is necessary because the number of fragments obtained from a given RNA during library preparation, and thus the read count for that transcript, is proportional to its length [61]. Only in the case of MMSEQ we have used the exponential of the reported means of the posterior distributions μ instead of RPKM (see above). However, these units are largely equivalent as they both control for sample size and feature length [141]. In cases where estimates were absent for individual transcripts, these were set to zero. For the comparisons of RPKM estimates with A-seq-2-based estimates (human and mouse), only those poly(A)-processing regions were considered that correspond to the ends of transcripts annotated in the GENCODE annotation sets (and vice versa). However, to account for the fact that only poly(A)-containing transcripts are efficiently captured by our sequencing library preparation protocols, we only considered transcripts which we presume could have been polyadenylated (annotated as either 'antisense', 'lincRNA', 'nonsense_mediated_decay', 'processed_pseudogene', 'processed_transcript', 'protein_coding', or 'retained_intron'). RPKM estimates for the remaining processing sites (25'393 and 17'183 for v19 human assembly version and M2 mouse assembly version, respectively) were then obtained by summing the RKPM values of the transcripts ending at individual poly(A)-processing regions. Similarly, we calculated gene-level expression estimates by summing the RPKM values of all transcripts (simulated data) or the TPM values of all processing regions (human and mouse data) annotated for each gene. Some of the benchmarked methods (Cufflinks, IsoEM, MMSEQ, RSEM, and rSeq) already provide gene-level estimates. However, for Cufflinks and MMSEQ these are not fully equivalent to the sums computed as described above. In the case of Cufflinks, this is apparently because of residual counts that could not be confidently assigned to any of the isoforms of a gene, since in the transcript output for that method ('isoforms.fpkm_tracking') there is reported for each gene an estimate that accounts for the difference between the sum of transcript isoform estimates and the gene expression estimates reported in a separate file ('genes.fpkm_tracking'). For MMSEQ, gene

level estimates are produced by aggregating the Markov chain Monte Carlo traces for the transcripts originating from a gene locus. Whenever gene-level estimates of expression were directly reported by a method, we have used these. As with transcripts, missing gene expression estimates were set to zero.

2.6.9 *Count-based gene-level estimates of expression*

Although our primary interest was to assess the accuracy of methods for isoform expression profiling, a lot of studies rather limit themselves to gene-level estimates of gene expression. The question then arises of how the methods that are used for obtaining gene-level estimates compare with those that are specifically designed for estimating isoform abundance but can be co-opted for the estimation of gene-level expression levels as well. One method for estimating gene-level expression is 'union exon'-based counting. To implement this method we have selected the exon entries from the GENCODE annotation files, grouped them by the ENSEMBL gene identifier, and merged overlapping exons for each gene. When analyzing human or mouse data, we have discarded the exons of transcripts that do not end in annotated poly(A)-processing regions or that are unlikely to be polyadenylated, analogous to the filtering that we applied to transcripts used in the benchmarking (see above). We then generated per-gene counts by intersecting the genomic alignments of the different datasets with the resulting 'pseudoexons', using the function summarizeOverlaps of the R/Bioconductor package GenomicAlignments [171] with options –ignoreStrand=FALSE, –mode='IntersectionStrict' and –interFeature=TRUE. While this procedure prevents double-counting of reads and is frequently applied in the context of gene counting in RNA-seq experiments, reads aligning to multiple genomic loci as well as those aligning to loci for which more than one feature is annotated are not considered. Additionally, many read alignments covering exon-exon-junctions are discarded because these exon-exon junctions are not part of the set of junctions between pseudo-exons. To appropriately handle such cases we implemented also a 'transcript'-based counting method as follows: We used the R/Bioconductor package rtracklayer [170] to convert the GENCODE-annotated exons of either all (in silico-generated data) or the filtered set of transcripts (human and mouse data; see above) to the BED12 / BED12+3 format, a tabular format able to encode gaps. We then intersected the genomic alignments for each dataset with the corresponding annotation file using bedtools mode intersect [57] such that overlaps were only reported if the entire read alignments, including the gaps that could correspond to introns, matched the transcript alignments on the sense strand (options -s and -f 1). The resulting overlaps were summarized, further distributing reads equally to all (possibly overlapping) annotated loci to which they aligned with the same edit distance. Thus, we first determined the number of genomic loci l for which overlaps were reported for a given read. For each

of these, we then added $\frac{1}{\sum_{i=1}^{l} g_i}$ to the total count of all genes that give rise to one or more transcripts from a locus i. For each library, the counts produced by each of these counting methods were then converted to RPKM by dividing by (1) the total number of reads that could be successfully aligned to the genome and (2) the total length (in nucleotides) of the 'union exons' (see above) of the considered transcripts, followed by multiplication by 1 billion.

2.6.10 *Evaluating the accuracy of gene/isoform abundance estimates*

We assessed the accuracy of the methods in terms of Spearman correlation coefficients between the known (simulated data) or independently estimated abundances (A-seq-2) and the abundances inferred with the individual methods. Depending on the type of data and analysis, we applied this procedure on the level of transcripts, poly(A)-processing regions, and/or genes, either considering all features or subsets thereof, grouped by common features (for example, expression ranges, structural). Where indicated, we have further computed the Pearson correlation coefficient and the root mean square error. In these cases, we have first set all expression levels (true or estimated) below 0.03125 (the log2 of which is -5) to that value and log2-transformed the resulting 'pseudocount'-adjusted values.

2.6.11 *Availability of supporting data*

Raw sequencing (RNA-seq and A-seq-2) and in silico-generated read files are available in the Sequence Read Archive (SRA) http://www.ncbi.nlm.nih.gov/sra repository under accession SRP051039 http://www.ncbi.nlm.nih.gov/sra/?term=SRP051039. As the SRA currently only supports the deposition of read alignments to genomic sequences, we have hosted the processed transcriptome alignment files, corresponding to the simulated/synthetic and experimental (RNA-seq) read libraries, on our companion website http://www.clipz.unibas.ch/benchmarking. The page further includes information on where to find the benchmarked methods, all source code - organized in well documented convenient wrappers that allow easy recreation of either the whole study or parts thereof - and an online analysis service where users can upload expression estimates inferred from the datasets used in this study and compare them to the methods (or their specific versions) assessed here.

2.7 ACKNOWLEDGEMENTS

We are grateful to Peter Glaus, Magnus Rattray, Antti Honkela (BitSeq), Wei Li (CEM), Cole Trapnell (Cufflinks), Ernest Turro (MMSEQ), Colin Dewey (RSEM), Hui Jiang (rSeq), Rob Patro (Sailfish), Sabah Kadri (Scripture), and Naoki Nariai (TIGAR2) for providing valuable assistance with this study. We apologize to the developers of transcript isoform quantification methods that were not included in this

survey because they could not be implemented, did not meet the requirements imposed by our study design or that we simply did not find in our initial method search. We thank Christoph Rodak for help with setting up the companion website and Manuel Belmadani for contributing scripts for the A-seq-2 analysis. This work was supported in part by a Sinergia grant from the Swiss National Science Foundation (CRSII3_127454), a Marie Curie Initial Training Network (project #607720, RNATRAIN), and a Starting Grant from the European Research Council (#310510, WHYMIR).

3

TERMINAL EXON CHARACTERIZATION WITH TECTOOL REVEALS AN ABUNDANCE OF CELL-SPECIFIC ISOFORMS

3.1 ABSTRACT

Sequencing of RNA 3′ ends has uncovered numerous sites that do not correspond to the termination sites of known transcripts. Through their 3′ untranslated regions, protein-coding RNAs interact with RNA-binding proteins and microRNAs, which regulate many properties, including RNA stability and subcellular localization. We developed the terminal exon characterization (TEC) tool (http://tectool.unibas.ch), which can be used with RNA-sequencing data from any species for which a genome annotation that includes sites of RNA cleavage and polyadenylation is available. We discovered hundreds of previously unknown isoforms and cell-type-specific terminal exons in human cells. Ribosome profiling data revealed that many of these isoforms were translated. By applying TECtool to single-cell sequencing data, we found that the newly identified isoforms were expressed in subpopulations of cells. Thus, TECtool enables the identification of previously unknown isoforms in well-studied cell systems and in rare cell types.

3.2 INTRODUCTION

Most eukaryotic transcripts undergo maturation through 3′-end cleavage and polyadenylation (CPA). The 3′ untranslated regions (3′ UTRs) of protein-coding messenger RNAs (mRNAs) interact with RNA-binding proteins (RBPs) [91] and microRNAs (miRNAs), which control diverse aspects of gene expression [172]. Global changes in 3′-UTR length have been observed during immune responses [173] and development [174], as well as in cancers [99]. It was initially thought that 3′-UTR shortening serves to counteract the repressive effect of miRNAs in proliferating cells [173] [175]. However, subsequent studies found largely similar decay rates of long and short 3′-UTR isoforms [176] [167], which left the role of changes in 3′-UTR length unclear. Evidence is accumulating that 3′-UTR-located sequence elements, particularly those that are uridine (U) rich, regulate many aspects of gene expression, from alternative polyadenylation in the nucleus to the subcellular localization of mRNAs and proteins in the cytoplasm [94] [177] [16].

In spite of many efforts to catalog human and mouse transcript isoforms [97] [15] [178] [179], sequencing of RNA 3′ ends continues to uncover novel polyadenylation (poly(A)) sites (PASs), many of which are outside of annotated exons [97] [15] [178] [179]. The presence of

well-characterized poly(A) signals indicates that these PASs are genuine [94], yet little is known about their regulation and functions. In a recent study [94], we identified 108,932 PASs in genomic regions annotated as introns in the GENCODE transcript annotation, version 19 (ref. [26]). In contrast to the more studied tandem PASs in 3′ UTRs, whose variable processing leads to changes in 3′-UTR length, the use of 'intronic' PASs can alter both the encoded protein isoforms and the 3′ UTRs, and thus the interactomes of the corresponding transcripts. To expand genome annotations with transcripts that end at currently intronic PASs, we developed a computational terminal-exon-characterization tool, TECtool.

3.3 RESULTS

3.3.1 *Prevalent RNA processing at intronic poly(A) sites*

A large proportion of PASs reproducibly identified from ~200 distinct human and mouse 3′-end sequencing samples were located in genomic regions currently annotated as intronic [94]. Representing up to ~10% of the PASs identified in individual tissues (unrelated to sequencing depth; Figure 3.1 A and Supplemetary Figure B.1 A), intronic PASs had canonically positioned poly(A) signals (~21 nucleotides (nt) upstream of PASs; Figure 3.1 B and Supplemetary Figure B.1 B) but a more specific tissue distribution than the PASs in annotated terminal exons (Figure 3.1 C and Supplemetary Figure B.1 C).

3.3.2 *TECtool identifies terminal exons from RNA-sequencing data*

3′-end sequencing data remain relatively scarce. However, public databases contain many RNA-seq datasets, from a wide range of cell types that provide evidence for as-yet-unannotated transcript isoforms (Figure 3.2 A and Supplementary Figure B.2). TECtool identifies terminal exons and transcript isoforms ending at intronic PASs (Figure 3.2 B, C). On the basis of alignments of RNA-seq reads resulting from single- or paired-end sequencing (Supplementary Figure B.3), TECtool trains a model (Supplementary Figure B.4 A) to distinguish terminal exons from internal exons and background regions, using a variety of features that reflect differences in the coverage of these regions by RNA-seq reads (Supplementary Figure B.5). It then uses the model to predict previously unknown terminal exons, corresponding transcripts, and their putative coding regions. TECtool can also be applied to data from unstranded protocols (e.g., Illumina TruSeq RNA v2). In this case, it does not predict terminal exons that overlap with annotated exons encoded on the opposite strand. To analyze data from single cells, where most transcripts are only sparsely covered by reads, we designed a TECtool workflow that initially pools the reads to infer novel transcripts, and then quantifies the abundance of these transcripts in individual cells (Supplementary Figure B.6).

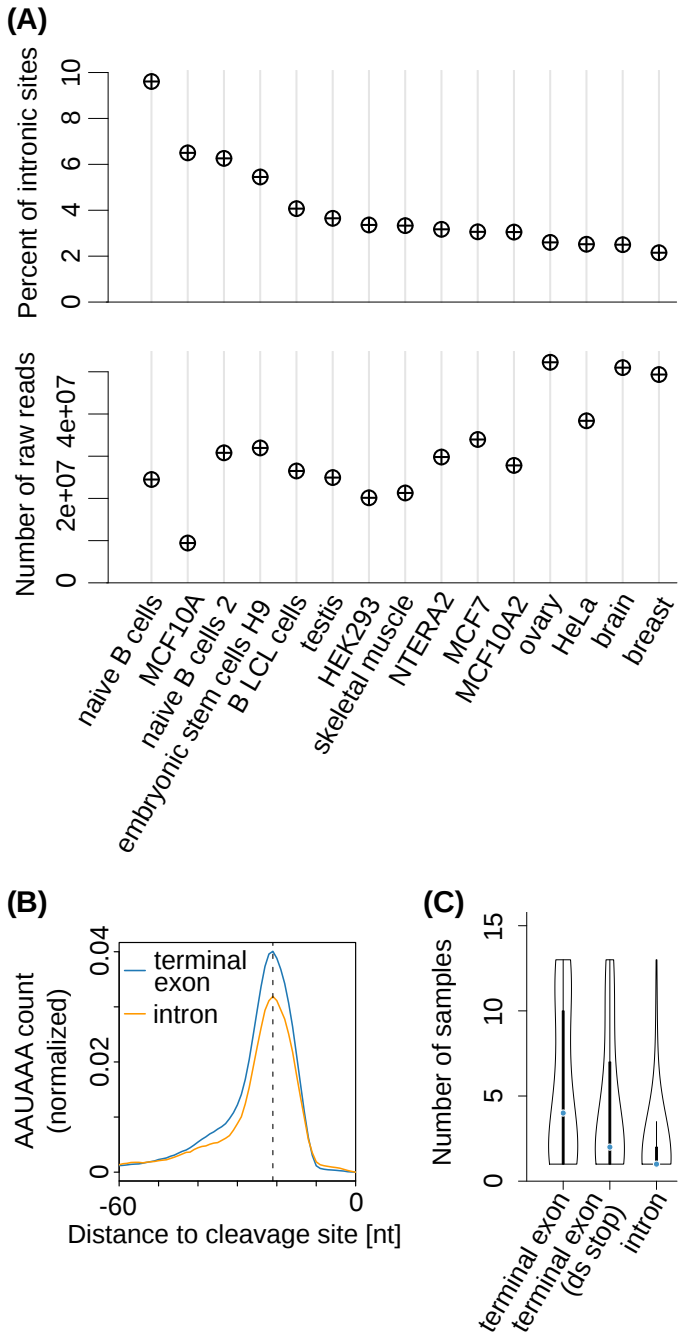


Figure 3.1: Cell-type-dependent use of intronic PASs. **(A)** Top, percentage of intronic PASs in individual samples obtained with the 3'-Seq protocol [180]. Bottom, corresponding sequencing depths. **(B)** Position-dependent frequency of the canonical poly(A) signal (AAUAAA; dashed line at -21 nt) upstream of intronic PASs (orange) and of PASs from annotated terminal exons (blue) from the study represented in A. **(C)** Distribution of the number of distinct samples in which individual PASs were observed, for PASs from terminal exons with no stop codon annotated downstream (terminal exon; 26,894 PASs), from annotated terminal exons located upstream of an annotated stop codon in the corresponding gene (terminal exon (ds stop); 3,430 PASs), and from genomic regions currently annotated as intronic (intron; 3,937 PASs). Black boxes indicate the interquartile range (IQR), blue dots indiate the median, whiskers corresponding to 1.5 times the IQR from the hinge, and densities extend to the most extreme values.

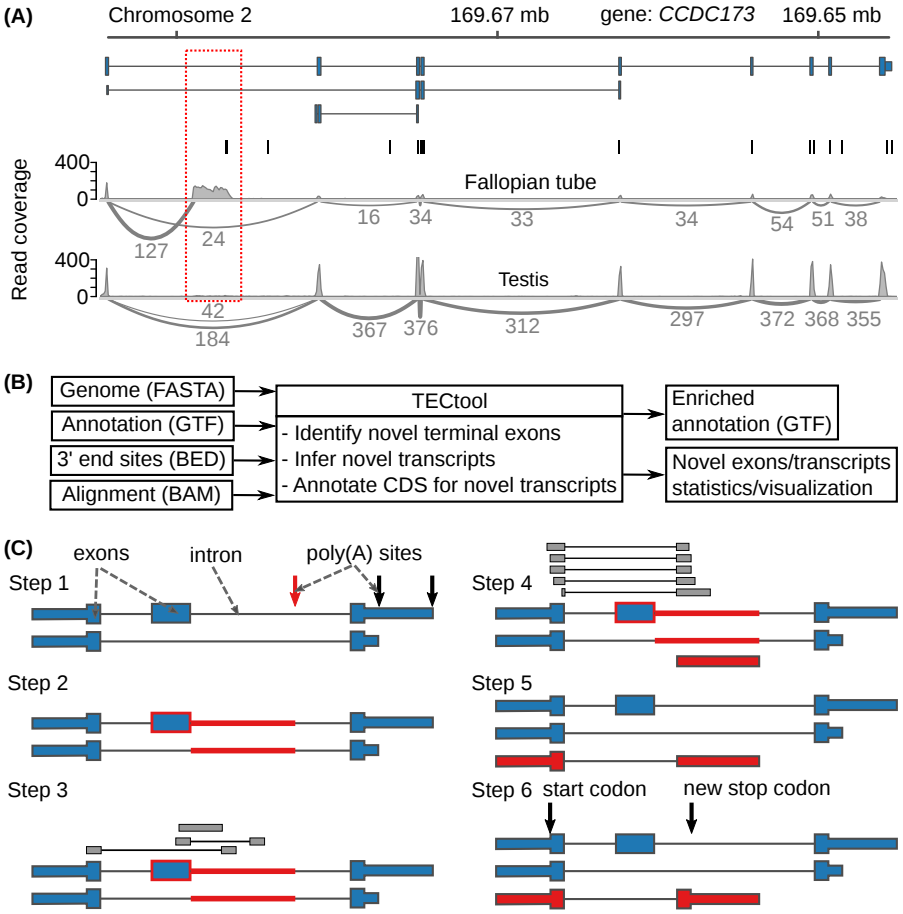


Figure 3.2: Example and model to identify novel 3'-UTR isoforms. **(A)** Sashimi plots [115] of RNA-seq reads mapped to a region in the locus for coiled-coil domain containing 173 (CCDC173), with the annotated ENSEMBL transcripts (blue), the PASs annotated in the PolyAsite atlas (vertical black lines; `http://polyasite.unibas.ch`), and densities of RNA-seq reads (gray) from fallopian tube and testis samples. The novel terminal exon is marked by the red dashed box, gray arcs indicate putative splice junctions, and numbers on the arcs indicate supporting reads (for clarity, only splice junctions supported by at least 10% of the maximum number of split reads between two exons in the genomic locus are shown; also see Supplementary Figure B.2 A). **(B)** Flow of data through TECtool (input and output file formats are indicated in parentheses). **(C)** Outline of the main computational steps. Step 1, selection of PASs located in regions that, with respect to the input annotation, are intronic (red arrow) and not exonic, intergenic, or antisense (black arrows). Step 2, identification of the feature region of the putative novel terminal exon (red line), extending from the intronic PAS up to the closest annotated exon upstream (blue box with red border). Step 3, identification of reads that map uniquely to the feature region. Step 4, definition of terminal exon boundaries (red box), given by a splice site at the 5' end, inferred from split reads, and the intronic PAS at the 3' end. Classification of putative terminal exons is done with a Bayes classifier. Step 5, the newly identified terminal exons are linked to upstream exons to which they were found to be spliced on the basis of split reads, to generate previously unknown isoforms. Step 6, prediction of protein-coding regions in newly identified transcripts.

3.3.3 *TECtool reproducibly and accurately identifies transcripts*

To evaluate TECtool, we took advantage of extensive datasets generated from human embryonic kidney (HEK) 293 cells. The 'support level' annotation of individual transcripts in ENSEMBL [25] provides a natural way to validate the tool, as we can determine whether isoforms that are predicted as novel relative to the annotation with the strongest experimental support (transcript support level (TSL) 1) are present in the annotation with more limited experimental evidence. From two biological replicates of RNA-seq in HEK293 cells [181], TECtool identified 327 and 337 terminal exons (510 in total and 154 in common) that were novel with respect to TSL1 annotation. We found that 321 of the 510 exons overlapped with terminal exons from the TSL1–5 annotation, and both annotated and novel transcripts had very reproducible expression in the two replicates (Supplementary Figure B.7 A). When we repeated the inference starting from the known TSL1–5 transcripts, we obtained 170 and 150 novel terminal exons in the two replicates (250 total, 70 common), which were similar in properties to transcripts identified starting from TSL1 annotation (Figure 3.3 A). These results show that TECtool is able to identify many previously unknown terminal exons even from a highly studied cell line such as HEK293.

Ribosome profiling data from HEK293 cells [182] revealed that the identified terminal exons had much higher translational efficiency than intronic sequences, but lower efficiency than already annotated terminal exons (Figure 3.3 B and Supplementary Figure B.7 B). The ribosome footprint density peaked around stop codons, whether already annotated or predicted in the novel terminal exon isoforms (Supplementary Figure B.7 C). These results indicate that TECtool-predicted isoforms are sufficiently stable to undergo translation.

The median lengths of TECtool-predicted terminal exons in the two HEK293 RNA-seq samples were 732 and 632 nt, respectively, which are longer than the median lengths of terminal exons predicted by StringTie [183] (380 and 412 nt, respectively) and Cufflinks [138] (199 and 232 nt, respectively), the two currently most accurate transcript reconstruction methods [67] (Figure 3.3 C). TECtool did not predict any exon shorter than 50 nt, in contrast to StringTie (3.5% and 3.9% of terminal exons in the two replicates, respectively) and especially Cufflinks (21.5% and 22.4%, respectively). This is a reflection of transcript reconstruction tools being largely unable to correctly determine transcript 3′ ends, where the coverage by RNA-seq reads is reduced. Consistent with accurate PAS assignment, only TECtool-predicted terminal exons had the canonical poly(A) signal (AAUAAA) at the expected position, ~21 nt upstream of PASs (Supplementary Figure B.7 D). In fact, only a minority of intronic terminal exons predicted by Cufflinks (32.4% and 31.4%) and StringTie (45.5% and 48.6%) had experimentally identified intronic PASs in the region +/-200 nt from their 3′ end (Figure 3.3 D). Even when we defined unique terminal exons solely by their splicing-determined 5′ end, TECtool made more reproducible predictions from replicate samples (40% of the union of

predicted exons were identified from both replicates, compared with 30% for StringTie and 18% for Cufflinks; Supplementary Figure B.7 E,F), whereas its predictions were largely not covered by the other tools (58% or 63% of its predicted novel exons). Thus, TECtool identifies with high reproducibility many exons not found by transcript reconstruction methods, with the unique advantage of accurate annotation of transcript 3′ ends. TECtool further predicts the coding region of novel transcripts, thus facilitating downstream analyses of encoded proteins.

In a recent study Lagarde et al.[184] sequenced samples from four tissues, in parallel on both short-read and long-read sequencing platforms, and their results allowed us to further validate TECtool-generated transcript models. Even though full-length RNA Capture Long Seq (CLS) primarily captures transcripts with high expression (Supplementary Figure B.8), ~8% of the novel transcripts predicted by TECtool from short-read sequencing were also identified by long-read sequencing (44, 5, 0, and 1 of the 464, 88, 63, and 20 novel transcripts predicted from testis, brain, heart, and liver samples, respectively). Thus, CLS validates highly expressed TECtool-predicted transcripts, and altogether our analysis shows that TECtool can substantially improve transcriptome annotation.

3.3.4 *TECtool identifies cell-type-specific isoforms*

Using an RNA-seq dataset that covers 32 human tissues [185], we identified hundreds of previously unknown terminal exons with TECtool, primarily from testis and bone marrow samples (Figure 3.4 A). This was not a mere reflection of the size of the libraries (Supplementary Figure B.9). Furthermore, many previously unknown isoforms were the most expressed transcripts of their corresponding genes (Supplementary Figure B.10), which indicates a special relevance of intronic PASs in these tissues.

3.3.5 *Previously unknown isoforms are expressed in subsets of single cells*

Single-cell RNA-seq allows one to assess whether a low average expression of a particular transcript results from 'transcriptional noise', affecting all cells, or from highly specific expression in rare cell types. By applying TECtool to a recently published single cell RNA-seq dataset of 201 T cells [186], we found that the distribution of expression levels for novel isoforms in individual cells was in the range of that of annotated isoforms. Once transcripts reached an average expression of 1–2 reads per million per cell (considering only reads that spliced into the 5′ splice site of the terminal exon), we started to detect them in multiple cells (Supplementary Figure B.11 A). However, multiple isoforms with distinct terminal exons were rarely present in a cell at the same time (Supplementary Figure B.11 B). Thus, rather than being coexpressed with the more abundant annotated isoforms, novel isoforms appeared to be expressed in subsets of cells, at a per-

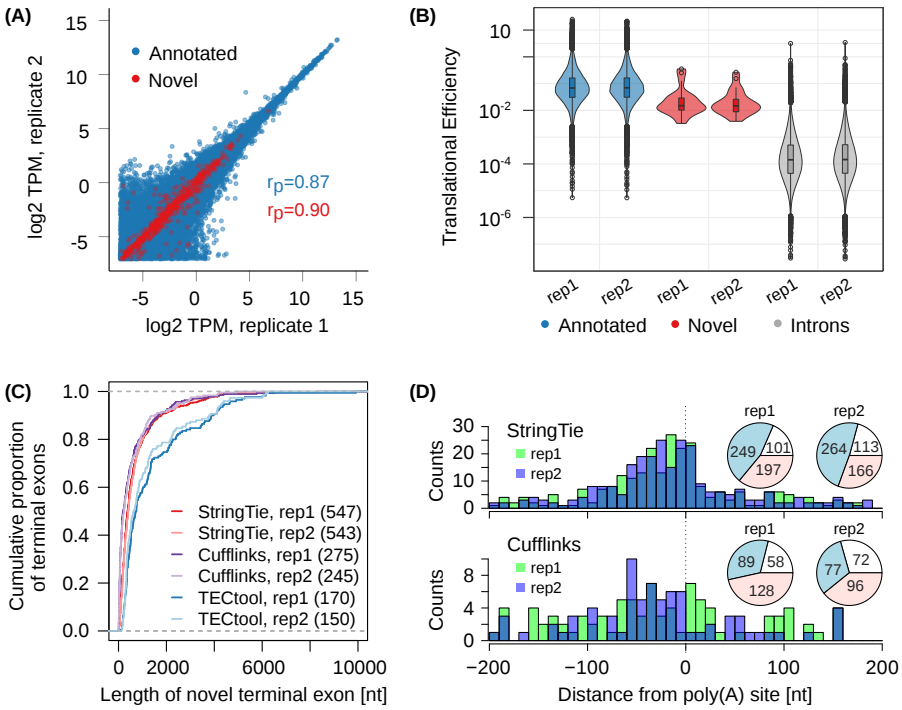


Figure 3.3: Evaluation of TECtool's performance. **(A)** Scatter plot of estimated expression levels of already annotated transcripts (ENSEMBL v87; TSL1-5, blue, 168,726 transcripts) and of transcripts ending at TECtool-identified terminal exons (red; 842 novel transcripts) in biological replicates of RNA-seq from HEK293 cells (rp indicates the corresponding Pearson correlation). TPM, transcripts per million. **(B)** Translational efficiencies computed for annotated terminal exons, novel terminal exons, and intronic regions (two-tailed t test P values for pairwise comparisons of regions based on TSL1-5: novel versus intron replicate 1 (rep1) , 2.1×10^{16}; replicate 2 (rep2), 5.4×10^{18}; annotated versus novel, rep1, 1.4×10^{-5}; rep2, 8.6×10^{-7}). The numbers of annotated and novel exons and of introns were 16,068, 24, and 64,455 in rep1, and 15,772, 25, and 63,932 in rep2. Boxes indicate the IQR, with the center line corresponding to the median; whiskers extend to the most extreme value within 1.5 times the IQR from the hinge, and outliers beyond this range are shown as individual points. **(C)** Cumulative distribution of the length of novel terminal exons identified by TECtool, StringTie, and Cufflinks in the two replicate RNA-seq datasets, relative to the TSL1-5 annotation. The number of novel terminal exons identified by each tool is indicated in parentheses. **(D)** Distance between experimentally determined PASs from the PolyAsite atlas [94] and the 3' ends of novel transcripts identified by StringTie (top) and Cufflinks (bottom). Pie charts show the number of 3' ends of novel transcripts that had an experimentally determined PAS within +/-200 nt (blue), had experimentally determined PASs farther away but in the same intron (red), or did not have any experimentally observed PASs in the respective intron (white).

cell level similar to that of annotated transcripts (Figure 3.4 B and (Supplementary Figure B.11 C, D). These results illustrate the potential of TECtool to improve the characterization of transcript isoforms expressed in individual cells, thereby enabling the characterization of rare cell types.

3.4 DISCUSSION

After the initial assembly of the human genome [2] [1], full-length RNAs and expressed sequence tags were used to annotate gene structures [26] [187]. However, many transcripts that are specific to cell types or conditions remain uncharacterized, even though targeted sequencing of RNA 3′ ends hints at their existence [94] [15]. Although analysis of RNA-seq data increasingly involves transcript reconstruction, the accuracy of the approach is limited by alignment errors, intron retention events, and 3′ -end bias when poly(A) selection is performed [67] [188]. Terminal exons are especially problematic, as the transcript coverage by RNA-seq reads decreases toward the 3′ end. We demonstrated that the accuracy of isoform annotation can be substantially improved by the incorporation of experimentally identified PASs in transcript reconstruction. The approach can be applied to any RNA-seq dataset from a species for which PASs have been mapped, including human and mouse (Supplementary Figure B.12). Similar to transcript reconstruction methods, TECtool relies on high-quality RNA-seq data from samples with minimal RNA degradation and little bias in coverage along transcripts. We obtained good results with samples for which transcript integrity scores [189] were greater than 0.8. TECtool also provides the option to analyze new datasets with a model build from samples with deep coverage and high RNA integrity, such as the HEK293 RNA-seq datasets that we used in this study, which permits analysis of samples with insufficient coverage and training examples.

Although third-generation sequencing technologies have made the full-length sequencing of RNAs more common, the capture of low-abundance transcripts remains very limited. By making use of extensive short-read sequencing data available from cell populations, and especially from single cells, TECtool supports the identification of even relatively rare transcripts. The tool is fully automated and easy to use. It does not require any customized input files or specific parameters, as it trains its own classifier on the basis of the input data.

3.5 METHODS

3.5.1 *Datasets*

Datasets used were downloaded either from `http://polyasite.unibas.ch`, or GEO data base [21], or Array Express data base [190]. Table 3.1 summarizes all the datasets.

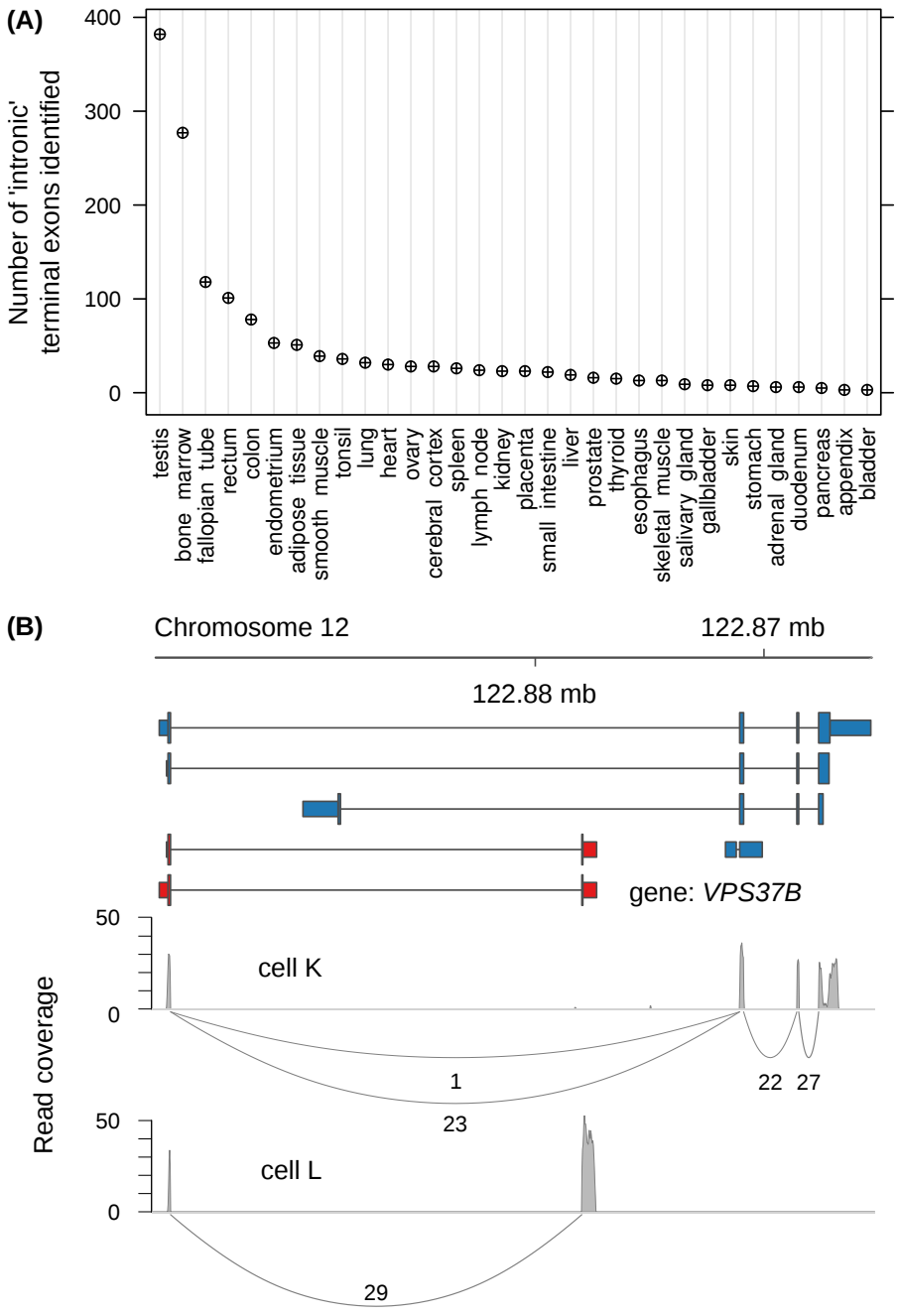


Figure 3.4: TECtool identifies previously unknown isoforms with cell-type-specific expression. **(A)** Number of previously unknown terminal exons identified by TECtool in at least one sample from the indicated tissues. **(B)** VPS37B gene locus with the ENSEMBL-annotated transcripts (blue), previously unknown transcripts predicted by TECtool (red), and Sashimi [115] plots of RNA-seq read densities (gray) from two single T cells (labeled as cells K and L, respectively).

Table 3.1: Datasets used for TECtool analysis

Dataset	Dataset reference	Downloaded from
3′ end sequencing	[94]	(http://polyasite.unibas.ch)
RNA-seq in HEK 293	[181]	GEO database [21]: GSE56010
Ribosome profiling in HEK 293	[182]	GEO database [21]: GSE73136
RNA-seq in tissues from Protein Atlas	[185]	ArrayExpress database [190]: E-MTAB-2836
RNA-seq and PacBio reads in 4 different tissues	[184]	GEO database [21]: GSE93848
Single-cell data	[186]	GEO database [21]: GSE85527
Mouse data	[191]	GEO database [21]: GSE52260

3.5.2 *Poly(A) sites*

The genome coordinates of the poly(A) sites from the recently published atlas [94] were converted to the GRCh38 genome assembly version with liftOver [192].

3.5.3 *Analysis of intronic poly(A) sites identified by 3′-end processing*

The locations of all poly(A) sites were associated with the set of transcripts of support levels 1-5 from the ENSEMBL gene annotation version 87 [187]. Pre-mRNA cleavage sites inferred from the samples that were part of two 3′ end sequencing studies, utilizing either the 3′-Seq [180] or the SAPAS [193] protocol, were intersected with the annotated PAS and sets of PAS expressed at the level of at least five reads per million in individual samples were identified. PAS from introns, terminal exons or terminal exons with downstream stop codon in the same gene were identified in the basis of the ENSEMBL annotation.

3.5.4 *TECtool*

TECtool is implemented as open source Python (version 3.4 and higher) software that can be obtained from http://tectool.unibas.ch. It depends on the packages HTSeq [55] (version 0.9.1), Bedtools [57] (version 2.26.0), Pybedtools [194] (version 0.7.10), pyfasta (version 0.5.2), numpy [195] (version 1.13), scipy [196] (version 0.19), scikit-learn [197] (version 0.19.0), pandas [198] (version 0.2) and progress (version 1.3).

3.5.4.1 *Inputs, outputs and user options*

TECtool requires the following inputs (Figure 3.2 B): (1) a file containing all chromosomes in fasta format, (2) a file with the corresponding annotation in ENSEMBL GTF format [187], (3) a file with genome coordinates of 3′ end processing sites (in BED format) and (4) a file containing spliced alignments of RNA-seq reads to the corresponding genome (in BAM format, sorted by coordinates and indexed). For human and mouse, downloadable files of poly(A) sites can be found on the website of the PolyAsite atlas http://polyasite.unibas.ch, [94]). The output of TECtool (Figure 3.2 B) is an augmented annotation file (in GTF format), that contains the input as well as the newly annotated transcripts. Additional files, summarizing the features of

annotated and newly identified exons, that are generated during the run, are also provided (in tab-delimited format). The tool requires that the sequencing direction be specified (as forward/unstranded) for the reads in the BAM file using the –sequencing_direction flag. Other implemented options allow the specification of the number of spliced reads required to support a novel exon, or whether to enforce the use of specific features in training the model and predicting new terminal exons. The tool can also be run with a user-specified, pre-trained model (TECtool options: –use_precalculated_training_set, –training_set_directory) that the user would need to obtain in a preliminary run with a dataset with good transcript coverage by reads. This may be useful when the coverage of annotated exons in the input RNA-seq data is low and therefore too few data are available to train an appropriate model.

3.5.4.2 *Selection of intronic PASs*

In a first step, TECtool uses the provided transcript and PAS annotations of the genome to select candidate intronic PASs. These are located within the loci of annotated genes, but outside of annotated exons. When the RNA-seq data used did not preserve strand information, TECtool discards PASs that are located in introns of genes that have other exons annotated on the complementary strand.

3.5.4.3 *Identification of candidate novel terminal exon*

For each intronic PAS, TECtool defines a 'feature' region that extends from the PAS to the closest upstream exon (Figure 3.2 C). The upstream exon is considered the 'reference' region. When the upstream exon has multiple possible 5' ends, the longest exon variant becomes the 'reference' region.

Uniquely mapping reads overlapping the feature region, either unspliced or mapping across splice junctions, with the 5' end in an exon upstream of the candidate intronic PAS and the 3' splice site within the feature region, are identified. When the number of such spliced reads surpasses a user-defined lower bound (default: five reads), a putative terminal exon is constructed extending from the 5' splice site of the spliced reads to the intronic PAS. Potential terminal exons that overlap with annotated exons of other genes are not considered.

3.5.4.4 *Collection of training exonic regions*

TECtool aims to classify the following:

1. Terminal exons: unique last exons of annotated transcripts, as defined in the provided annotation file, not including exons that overlap with other exons or that do not have the (user-)defined minimum number of splice-in reads (default: five reads).

2. Internal exons: exons that are neither the first nor the last exon of an annotated transcript, do not overlap with any other exon, and have the (user-)defined minimum number of splice-in reads.

3. 'Background' regions: annotated terminal exons that do not overlap with other exons but have less than the (user-)defined minimum number of splice-in reads.

3.5.4.5 *Feature computation*

For each exonic region in the training set (object), TECtool computes the following features Supplementary Figure B.3 B-H:

- Splicing-in-boundary/all: counts of reads that splice from an upstream region into the 5′ boundary/anywhere within the entire length of the object.
- Splicing-out-boundary/all: counts of reads that splice from the 3′ boundary/anywhere within the entire length of the object to a downstream region.
- Crossing-in/out-boundary: counts of unspliced reads overlapping the 5′/3′ boundary of the object.
- Unspliced-within-boundaries: counts of unspliced reads that are contained in the object.
- Reads-within-gene-loci: number of reads that map within gene loci.
- Union-exon-length: length of the union exons of the gene.

TECtool then calculates (Supplementary Figure B.5):

- Reads-out versus reads-in ratio: the ratio of reads splicing out or crossing the 3′ boundary of the object and reads splicing in or crossing the 5′ boundary of the object.
- Normalized region expression: ratio between the expression of the object (per kilobase, including splicing-in/out-all, crossing-in/out-boundary, and unspliced-within-boundary reads) and the expression of the corresponding gene (per kilobase, reads-within-gene-loci divided by length of union-exons).
- Object length.
- Entropy efficiency: a measure of the 'uniformity' of read coverage along the object, defined as the Shannon entropy of read coverage per position divided by the maximum value it can take based on the object length,

$$EE(x) = -\frac{\sum_{i=1}^{n} p(x_i)\log(p(x_i))}{\log(n)} \tag{3.1}$$

where n represents the length of the object and $p(x_i)$ is the coverage at position i divided by the total coverage along the object,

$$p(x_i) = \frac{x_i}{\sum_{j=1}^{n} x_j} \tag{3.2}$$

EE(x) takes values between 0 and 1.

- Relative positions of 5% and 95% quantile coverage: where the cumulative distribution of read coverage along the object reaches 5% and 95%.
- Splicing-in-all versus 5′ end expression: ratio between the number of reads splicing into the object (see "Splicing-in-all" above) and the mean coverage per position over the first 10 nt of the object.
- Splicing-out-all versus 3′ end expression: ratio between the number of reads splicing out from the object (see "Splicing-out-all" above) and the mean coverage per position over the last 10 nt of the object.
- Crossing-in versus 5′ end expression: ratio between the number of reads overlapping the 5′ boundary of the object (see "Crossing-in-boundary" above) and the mean coverage per position over the first 10 nt of the object.
- Crossing-out versus 3′ end expression: ratio between the number of reads overlapping the 3′ boundary of the object (see "Crossing-out-boundary" above) and the mean coverage per position over the last 10 nt of the object.
- Splicing-in-boundary versus 5′ end expression: ratio between the number of reads splicing into the object (see "Splicing-in-boundary" above) and the mean coverage per position over the first 10 nt of the region.
- Splicing-out-boundary versus 3′ end expression: ratio between the number of reads splicing out from the object (see "Splicing-out-boundary" above) and the mean coverage per position over the last 10 nt of the region.
- Splicing-in-boundary versus Splicing-in-all: ratio between the number of reads splicing into the 5′ boundary of the object (see "Splicing-in-boundary" above) and the number of reads splicing into the object (see "Splicing-in-all" above).

3.5.5 *Classifier training and prediction of novel terminal exons*

TECtool samples randomly 20% of the training data for validation, and approximates the distributions of all features described above for each region type in the remaining 80% of the training data (Supplementary Figure B.4 A) using kernel density estimation (KDE). We chose an exponential kernel function to better approximate drops at the boundary of the empirical distributions. Under the assumption of uncorrelated features, the KDEs represent posterior probabilities to use in the Bayes classifier. As samples generated with different sequencing protocols typically have different coverage patterns along genes, the features that best distinguish exon types may change from

sample to sample. Thus, TECtool uses a forward greedy feature selection, incrementally and greedily adding features that increase the performance (F1-score t value > 1.37) on the validation data, starting from a core set of features (entropy efficiency and reads-out versus reads-in ratio). To increase the stability of the model, TECtool trains classifiers on ten randomized subsets of the training data (1,000 objects in each class, or the entire set if smaller than 1,000), and then uses each of them to evaluate each candidate terminal exon (Supplementary Figure B.4 A). The average probabilities that the candidate exon will be terminal, internal, or background, computed over the ten classifiers, determine the category to which the candidate exon is assigned. When there are multiple putative terminal exons with the same 5′ splice site but different PASs, only the exon with the highest probability of being terminal is reported in the final GTF.

3.5.6 *Novel transcripts and CDS annotation*

Having identified putative terminal exons, TECtool constructs putative novel transcripts, starting from annotated transcripts that contain an exon that splices to the novel terminal exon. These transcripts (which we call root transcripts) and upstream exons are identified on the basis of spliced reads.

In the final step, TECtool annotates the putative protein-coding region in the newly annotated transcripts. When the root transcript is protein coding, TECtool uses the already annotated start codon and searches for the first in-frame stop codon. If it is found, the novel transcript is annotated as protein coding. When the root transcript has no annotated start codon or when no in-frame stop codon is found, the transcript is classified as noncoding.

3.5.7 *Automated analysis of RNA-seq datasets with TECtool*

We implemented automated TECtool analyses of standard RNA-seq (Supplementary Figure B.3 A), as well as single-cell RNA-seq data (Supplementary Figure B.6). The analysis flows are implemented in the snakemake framework [30], and the parameters for each type of analysis are specified in a corresponding configuration file. The single-cell sequencing data pose the challenge of relative low and highly nonuniform coverage for most genes. Therefore, we initially pool the data from all cells in a sample to identify the terminal exons, which we then quantify in individual cells with a method for transcript isoform quantification.

3.5.8 *Analysis of mouse RNA-sequencing data*

To demonstrate the generality of the tool, we also applied it to RNA-seq data from a time series of mouse T cell activation (accession GSE52260). A summary of the results for individual time points is

shown in Supplementary Figure B.12, together with genome browser screenshots for two individual examples.

3.5.9 *Analysis of novel transcript expression in 32 human tissues*

We analyzed the mRNA-seq data generated for 32 human tissues [185] with TECtool. We merged the enriched annotation files corresponding to replicate samples from the same tissue, to construct tissue-specific annotation files. Estimates of transcript and gene expression levels in each tissue were obtained with Salmon [199].

3.5.10 *Visualization of read densities*

Sashimi plots [115] were generated with a custom script that is based on the following R libraries: Gviz [114], rtracklayer [170] and GenomicFeatures [171].

3.5.11 *Statistics*

For comparison of translation efficiency, we used two-tailed t tests, not treating the two variances as equal. The number of cases and P values are given in the legend of Figure 3.3.

3.5.11.1 *Analysis of single end RNA-seq data*

For single-end, bulk RNA-seq (Supplementary Figure B.3 A), TECtool first generates the required directories, optionally trims the 3′ adapters using cutadapt [38] (version 1.13)

```
cutadapt \
--adapter {3' adapter} \
--error-rate {error rate} \
--minimum-length {minimum read length} \
--overlap {overlap} \
{input reads} | gzip > {output reads}
```

and indexes the genome with the STAR software [51] (version 2.5.3a)

```
STAR \
--runMode genomeGenerate \
--sjdbOverhang {read length} \
--genomeDir {genome dir} \
--genomeFastaFiles {genome fasta file} \
--runThreadN {number of threads} \
--sjdbGTFfile {annotation file}
```

The reads are mapped to the genome with the STAR aligner

```
STAR \
--runMode alignReads \
--twopassMode Basic \
--runThreadN {number of threads} \
--genomeDir {genome dir} \
--sjdbGTFfile {annotation file} \
--readFilesIn {sample fastq file} \
```

```
--readFilesCommand zcat \
--outFileNamePrefix {prefix} \
--outSAMtype BAM Unsorted
```

Alignment files (in BAM format) are then sorted

```
samtools sort \
-@ {number of threads} \
{input bam} > {output bam}
```

and indexed

```
samtools index {input bam}
```

using samtools [45] (version 1.3). The sorted alignment file and appropriate input options are provided to TECtool

```
tectool \
--annotation {input annotation file} \
--polyasites {input polya sites} \
--bam {input alignment file} \
--sequencing_direction {sequencing direction option} \
--genome {genome fasta file} \
--minimum_spliced_reads_for_cryptic_exon_start_site 5 \
--output_dir {output directory}
```

to identify novel terminal exons and output an enriched annotation file (in gtf format). The gtf files from different replicates are then merged

```
tectool_add_novel_transcripts_to_gtf_file \
--list_of_gtf_files {list of gtf files} \
--out-dir {output directory}
```

into a single gtf file with a custom TECtool script. A file (fasta format) of transcript sequences is generated based on the annotation file with the gffread script from the cufflinks package [138] (version 2.2.1)

```
gffread \
{merged annotation file} \
-g {genome fasta file} \
-w {transcripts fasta file}
```

Finally, the transcriptome is indexed with Salmon [199] (v0.9.1)

```
salmon index \
--transcripts {input transcript sequences} \
--index {output index} \
--kmerLen {kmer length} \
--keepDuplicates \
--threads {number of threads}
```

and the expression levels of transcripts in each replicate are quantified

```
salmon quant \
--index {input index} \
--libType {library type} \
--unmatedReads {input reads} \
--seqBias \
```

```
--geneMap {merged annotation file} \
--fldMean {mean fragment length} \
--fldSD {standard deviation of fragment length} \
--useVBOpt \
--threads {number of threads} \
--output {output directory}
```

We used Salmon for isoform quantification (the successor of Sailfish [145]), because we found Sailfish to perform well relative to other methods for isoform quantification [200]. Similar results for Salmon were reported by others [201]. Other methods for transcript quantification can also be used. The output of this final step consists of transcript and gene expression estimates (in TPM).

3.5.11.2 *Analysis of paired-end RNA-seq data*

For paired-end bulk RNA-seq (Supplementary Figure B.3 A) TECtool first selects one of the two mates depending on a user-set parameter and then continues as described above. However, adapter trimming

```
cutadapt \
-a {3' adapter mate 1} \
-A {3' adapter mate 2} \
--error-rate {error rate} \
--minimum-length {minimum read length} \
--overlap {overlap} \
-o {mate 1 output reads} \
-p {mate 2 output reads} \
{mate 1 input reads} {mate 2 input reads}
```

and the transcript expression estimation

```
salmon quant \
--index {input index} \
--libType {library type} \
-1 {mate 1 input reads} \
-2 {mate 2 input reads} \
--seqBias \
--geneMap {merged annotation file} \
--useVBOpt \
--threads {number of threads} \
--output {output directory}
```

is performed in paired-end and not single-end mode.

3.5.11.3 *Analysis of single cell sequencing data*

Single cell sequencing data (Supplementary Figure B.6) poses the additional challenge that the coverage of individual genes is generally sparse and non-uniform. Thus, after the necessary directories are created, the genome is indexed with STAR and 3' adapters are trimmed with cutadapt, the read files corresponding to individual cells are concatenated, the reads are mapped to the genome with STAR and the alignment file is sorted and indexed with samtools. PCR duplicates are removed with samtools

```
samtools rmdup \
```

```
-s {input alignment file} \
{output alignment file}
```

and the resulting alignment file is provided as input to TECtool to identify novel exons and transcripts. Reads from individual cells are mapped to the genome with STAR and sorted with samtools. PCR duplicates are also removed from the alignment using samtools and then indexed. To quantify expression of transcripts that indeed include specific terminal exons, in spite of the sparse coverage of genes by reads in individual cells, we estimated the expression of novel and annotated transcripts as the number of split reads that fall in the 5′ splice junction of the respective exons (novel or annotated terminal exons that do not overlap with annotated internal exons).

3.5.12 *Analysis of TECtool running time*

Data from a time series of mouse T cell activation (accession GSE52260) were merged and mapped to the genome with STAR. After sorting the alignment file and removal of PCR duplicates with samtools, we kept only primary alignments

```
samtools view \
-F 0x100 \
-bS {input alignment file} \
> {output alignment file}
```

and generated subsets representing 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90% of the total number of reads

```
samtools view \
-s {subset fraction size} \
-b {input alignment file} \
> {output alignment file}
```

with samtools. We applied TECtool to each of the datasets, using the bash command time to obtain the running time. We repeated the procedure 10 times for each data set size to obtain averages and standard deviations of the running time for each data set size. All samples were run in a single core with 64GB of memory. The results are shown in Supplementary Figure B.4 B.

3.5.13 *Analysis of ribosome profiling data*

For the analysis of translation, we only considered novel terminal exons containing a stop codon. We mapped ribosome-protected reads to the genome with STAR and the parameters used for bulk RNA-seq data. We counted mapped reads with HTSeq [55], constructed profiles of ribosome footprints around the stop codon and estimate the density of ribosomes over the terminal exons. Estimates of transcript abundance (transcripts per million, TPM) from Salmon [199] were used to normalize Ribo-seq read densities (expressed in reads per kilobase per million, RPKM). The profile of ribosome footprints in intronic regions have been normalized to the expression level of the most expressed isoform of that gene (TPM), inferred with Salmon.

3.5.14 *Transcript and terminal exon reconstruction with StringTie*

StringTie [183] (version 1.3.3) has been found most accurate among transcript reconstruction methods in a recent benchmarking study [67]. We mapped RNA-seq reads obtained in a previous study [181] to the genome in paired end mode with the STAR aligner

```
STAR \
--runMode alignReads \
--twopassMode Basic \
--runThreadN {number of threads} \
--genomeDir {genome dir} \
--sjdbGTFfile {annotation file} \
--readFilesIn {sample mate 1 fastq file} {sample mate 2 fastq
    file}  \
--readFilesCommand zcat \
--outFileNamePrefix {prefix} \
--outSAMtype BAM Unsorted \
--outSAMstrandField intronMotif
```

After sorting and indexing the alignments with samtools, we ran StringTie with the following command

```
stringtie \
{input alignment file} \
-G {annotation file} \
-o  {output annotation file} \
{library type} \
-p {threads}
```

to generate gtf files with the new annotation. Finally, we extracted novel terminal exons that were located in introns relative to the support level 1-5 transcript annotation of the genome.

3.5.14.1 *Transcript and terminal exon reconstruction with Cufflinks*

Cufflinks [138] (version 2.2.1) is a second transcript reconstruction method with relatively good performance in a recent benchmarking [67]. We used the alignments of reads to genome obtained as described for Stringtie, and after sorting and indexing with samtools we ran Cufflinks with the following options

```
cufflinks \
--num-threads {number of threads} \
-g {annotation file} \
--library-type {library type} \
-o {output directory} {input alignment file}
```

obtaining a gtf file with novel and known transcripts for each sample. From transcripts with at least two exons, we then extracted terminal exons that were located in introns relative to the TSL1-5 annotation, using custom scripts and bedtools.

3.5.15 *Parallel analysis of long and short read data*

We used cutadapt [38] to trim 5′ and 3′ adapters, polyA and polyT stretches from PACbio reads, with the following commands:

```
cutadapt \
--front {5' adapter} \
--error-rate {error rate} \
--minimum-length {minimum read length} \
--overlap {overlap} \
{input reads} | gzip > {output reads}
```

and

```
cutadapt \
--adapter {3' adapter} \
--error-rate {error rate} \
--minimum-length {minimum read length} \
--overlap {overlap} \
{input reads} | gzip > {output reads}
```

We indexed the genome with the STAR [51] software (version 2.5.3a)

```
STARlong \
--runMode genomeGenerate \
--sjdbOverhang {read length} \
--genomeDir {genome dir} \
--genomeFastaFiles {genome fasta file} \
--runThreadN {number of threads} \
--sjdbGTFfile {annotation file})
```

and mapped the long, PACbio reads to the genome with the STAR-long version of the STAR aligner

```
STARlong \
--runMode alignReads \
--outFilterMultimapScoreRange 20 \
--outFilterScoreMinOverLread 0 \
--outFilterMatchNminOverLread 0.5 \
--outFilterMismatchNmax 1000 \
--winAnchorMultimapNmax 200 \
--seedSearchStartLmax 50 \
--seedPerReadNmax 100000 \
--seedPerWindowNmax 100 \
--alignTranscriptsPerReadNmax 100000  \
--alignTranscriptsPerWindowNmax 10000 \
--genomeSAsparseD 4 \
--outSAMunmapped Within \
--runThreadN {threads} \
--genomeDir {genome dir} \
--sjdbGTFfile {annotation file} \
--readFilesIn {input reads} \
--readFilesCommand zcat \
--outFileNamePrefix {prefix} \
--outSAMtype BAM Unsorted
```

We sorted

```
samtools sort \
-@ {number of threads} \
{input bam} > {output bam})
```

and indexed

```
samtools index {input bam}
```

the alignments with samtools [45]. After trimming the soft clipped parts of the mapped sequences,

```
samtools view -h {input alignment file in bam format} \
| awk 'BEGIN { OFS="\t"} { \
  split($6,C,/[0-9]*/); \
  split($6,L,/[SMDIN]/); \
  if (C[2]=="S") {
    $10=substr($10,L[1]+1); \
    $11=substr($11,L[1]+1); \
  }; \
  if (C[length(C)]=="S") {
    L1=length($10)-L[length(L)-1]; \
    $10=substr($10,1,L1); \
    $11=substr($11,1,L1); \
  }; \
  gsub(/[0-9]*S/,"",$6); \
  print \
  }' \
  | samtools view -bS - > {output alignment file in bam format}
```

we extracted transcript coordinates from the alignment file with bedtools [57] (version 2.26.0)

```
bedtools bamtobed \
-i {alignment file} \
-bed12 \
-splitD > {bed12 coordinates file}
```

to generate a bed12 file, and the extracted transcript sequences from the genome

```
bedtools getfasta \
-split \
-s \
-name \
-fi {genome sequence} \
-bed {transcript coordinates} \
-fo {transcript sequences}
```

In parallel we generated enriched annotation files by applying TEC-tool to the short read data also generated for the respective samples. We then mapped the PACbio-sequenced transcripts to annotated and novel transcripts with blast [202], first building the index of annotated and novel transcripts

```
makeblastdb \
-in {transcript sequences} \
-dbtype nucl \
-out {Blast database name}
```

and then running BLAST+ [203] (version 2.6.0)

```
blastn \
-num_threads {threads} \
-db {params.db_prefix} \
-query {PACbio extracted transcripts} \
-outfmt "6 qseqid sseqid pident qlen length slen mismatch gapopen
    evalue bitscore" \
-out {blast result}
```

From the blast output files we kept only transcripts that did not contain any internal gaps (gapopen=0), for which the alignment length (length) did not differ by more than 40 nts from either query (qlen) and target (slen) sequence lengths. This allows only small differences in the initiation/termination sites, but not incorrect assignment of splice variants.

3.6 ACKNOWLEDGEMENTS

We thank M. Jacquot and the members of sciCore for help with the infrastructure that we used to process the data. This work was supported by the Swiss National Science Foundation (grants 31003A_170216 (M.Z.) and 51NF40_141735 (National Center for Competence in Research 'RNA & Disease'; to the NCCR consortium)) and the Marie Curie Initial Training Network (project #607720, RNATRAIN; M.Z.).

4

DISCUSSION

Hundreds of thousands HTS data sets are available in public repositories such as GEO or SRA, and much more are expected to be sequenced and deposited in the near future. Most of them originate from technologies that generate short reads, such as Illumina [204]. These data are quite heterogeneous, since they were obtained from a variety of organisms, cell types, experimental protocols and instruments. Nevertheless, they reflect a large community effort in understanding biological processes and it would be highly desirable to make use of these data beyond the initial studies. In the work described in this thesis, I aimed to improve the characterization of gene expression by further exploiting existing high throughput data sets (mainly RNA-seq and 3'end sequencing). Towards this goal, I have developed a live tool for studying the performance of methods that quantify the abundance of transcript isoforms and I have developed a novel tool to uncover terminal exon isoforms.

In chapter 2 I described our strategy to benchmark programs that quantify isoform abundance. In contrast to most studies, that use for this purpose simulated RNA-seq reads, we have also attempted to quantify equivalence classes of transcripts (those that share the poly(A) site) by an independent experimental method, namely by mRNA 3' end sequencing. We applied this approach to human Jurkat and murine NIH/3T3 cells, from which we prepared and sequenced parallel samples by RNA-seq and 3' end sequencing (A-seq2 protocol). We felt that this was important, because RNA sequencing suffers from a variety of biases introduced by the different experimental steps that are not easy to simulate. We found that most programs for isoform quantification performed reasonably well, although a few 'winners' were apparent. Interestingly, all of these methods were more accurate in estimating gene level expression (by summing expression of transcripts associated with the gene) than widely used count-based approaches that handle poorly isoforms with different lengths. The main difference between the quantification methods lied not in their accuracy but rather in the amount of resources that they required (runtime and memory). This is important, because, as the size of the data sets continues to increase, only few current methods will continue to be practically usable. Importantly, we found that the transcripts and isoforms that have relatively low abundance are poorly quantified by essentially all methods, indicating that this is an area where substantially more work needs to be invested, computationally, but also experimentally.

In chapter 3 I described a novel computational tool (TECtool) to identify transcripts that contain unannotated terminal exons based on RNA-seq data and poly(A) sites obtained from 3' end sequencing experiments. The method is relevant because ¼ of the poly(A)

sites identified by 3′ end sequencing fall in regions annotated as introns. Transcript reconstruction methods are insufficiently accurate in detecting these exons, presumably because they have to be very stringent in separating spurious read coverage from expressed exons. This is is less of a problem for internal exons, for which reads supporting both the 5′ and 3′ splice sites are expected. In contrast, for terminal exons, the 3′ end is imprecisely defined and the coverage immediately upstream of the poly(A) site is typically poor, as very short fragments from the end of the transcripts are lost during the size selection step of the experimental protocol. With TECtool we found many novel tissue-specific transcripts. Tissues like bone marrow and testis are enriched in novel transcripts, but their functions are at the moment poorly understood. The enriched annotation that we provide opens the possibility to improve the quantification of expression of these novel transcripts across experimental conditions, to gain insights into the regulation of their expression and in the processes in which they are involved. Ribo-seq data indicate that many of the newly identified transcripts can be translated. With TECtool we could also explore the expression of terminal exon isoforms in single cells. Here we have noticed that transcripts that have low expression at the population level are not expressed at uniformly low levels in single cells, but rather they are well expressed in a smaller proportion of cells than transcripts with higher average expression. This is perhaps not unexpected, as transcription occurs in bursts, and it is possible that the transcripts of a burst experience a similar processing environment, which leads to their being more similarly processed than transcripts that expressed from the same gene but at very different times or in different cells. This seems an interesting line of further investigation based on single cell data. Finally, while technologies for full-length cDNA or RNA sequencing are available, we found that TECtool can identify more novel transcripts based on short read data, simply because the throughput of technologies generating long reads (such as PACBio) is much lower.

With my computer science background, I made specific efforts to make the tools available to the public and enforce the reproducibility of the analyses [205]. The code to generate all results and figures of chapter 2 is publicly available in a repository `https://github.com/zavolanlab/IsoformQuantificationBenchmarking`. Additionally, I created a supplementary website where developers of new methods can upload estimates isoform expression generated by their tools based on the experimental and synthetic data that we made available, to compare the results with the results from the approaches we assessed already. Our idea of a live site for benchmarking computational methods was also adopted by another group, who also used a subset of the data to provide the community with a shiny application for comparing binary classification methods [206]. The TECtool and three additional repositories with pipelines for processing single-end, paired-end, and single-cell RNA-seq data can also be obtained from `http://tectool.unibas.ch/`. and a supplementary website with further information and data is also available. One of directions in which

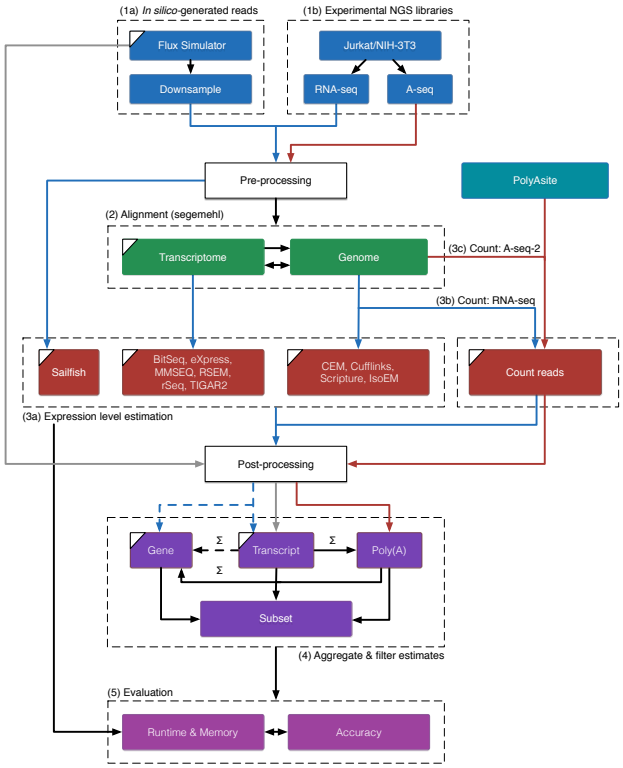
I am currently working is in automating to a very large extent the processing of HTS data. Specifically, I am developing a web server that will users to upload raw HTS data, process them with the methods described in these thesis and return publication-ready plots. As data processing has become an important bottleneck also for molecular biologists, this type of service could have an important impact on experimental research. I am especially interested in exploring the use of these tools in analyzing patient transcriptome data, which will likely be generated soon in personalized health projects.

I see a lot of potential in developing the TECtool further. Currently, TECtool processes every sample independently and cannot take advantage of replicates. In the future, the method can be improved by incorporating the variability between replicates in the model that scores putative terminal exons. TECtool is built explicitly for identifying transcripts with novel 3′ terminal exons. However, It is relatively easy to adapt the method to identify alternative 5′ exons starting from CAGE/promoter data instead of 3′ end sequencing data/poly(A) sites. Moreover, the tool can be further expanded to perform not only exon/transcript annotation but also differential expression of the identified terminal exons or transcripts. Many of the features and tools developed in these chapters can be used to build new methods to study splicing or RNA processing in general. For example, one could apply features used to identify novel terminal exons to improve the quantification of internal exon usage, which is currently done commonly through the percent spliced in score. Furthermore, one can also attempt to learn about and identify specific exons or events that current methods are not able to quantify, such as micro-exons or small variations at the 3′ splice site (so-called NAGNAG splicing). Ideally, a tool that can identify novel intermediate, start and terminal exons using RNA-seq, CAGE and 3′ end sequencing data, reconstruct transcript forms, quantify their expression and estimate the inclusion of specific events would be a great aid towards the effort of annotating and characterizing the transcriptome of different organisms and cells.

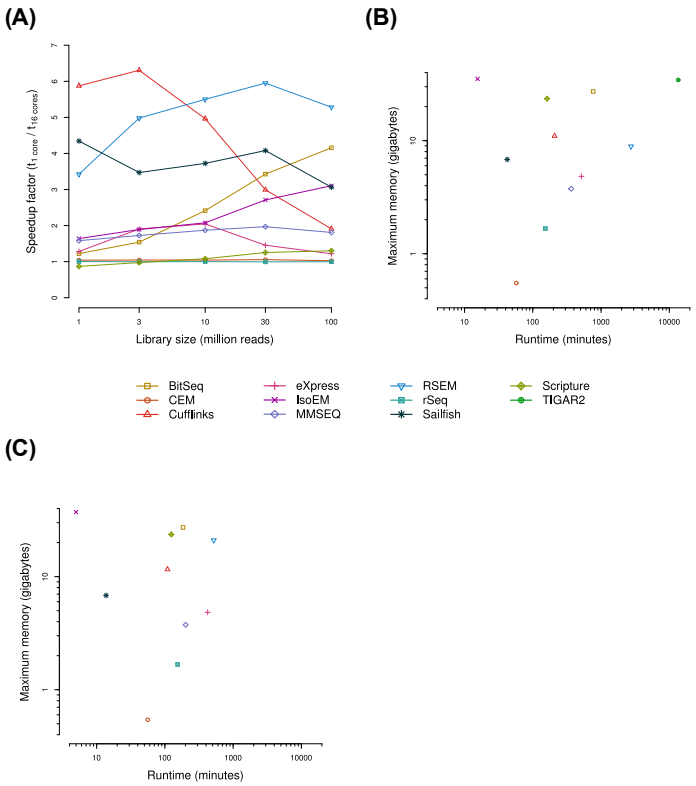
A

BENCHMARKING SUPPLEMENTS

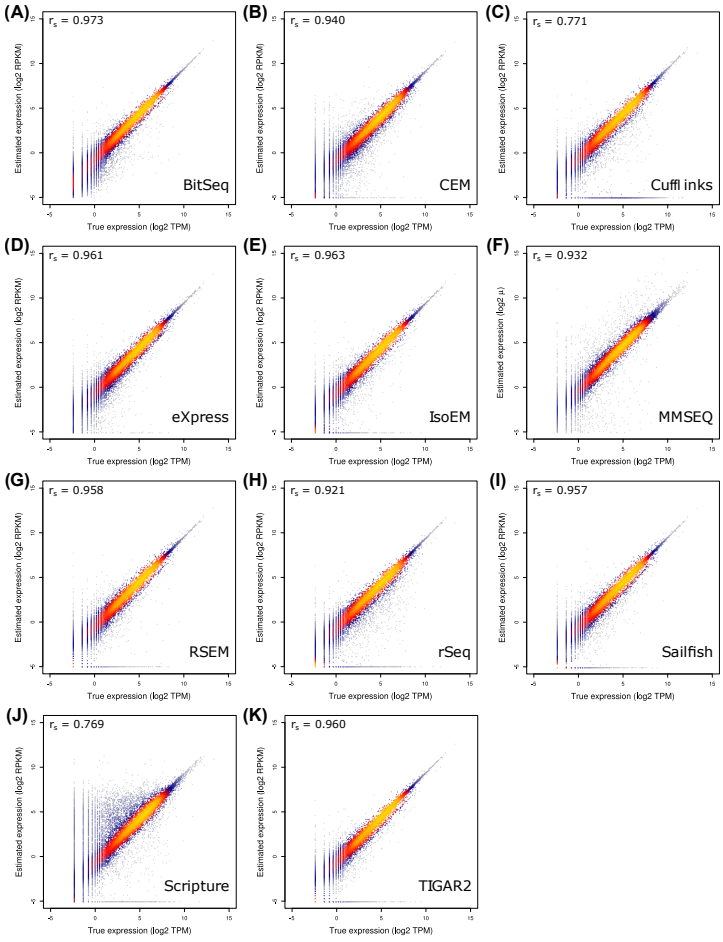
Supplementary material to chapter 2.



Supplementary Figure A.1: Overview of the study design. Sequencing data (blue boxes; 1) were generated synthetically (Flux Simulator; left side) or experimentally (right side) from human or mouse cells, following either a regular RNA-seq (blue arrows) or an A-seq-2 3′ end sequencing protocol (red arrows). 3′ adapters (if present) and poly(A)-tails were removed from read sequences ('pre-processing'), and the trimmed reads were then aligned against both the genome and the transcriptome (green boxes; 2). Genome alignments were supplemented with read alignments covering splice junctions by converting transcriptome alignments to genome coordinates. Genome and transcriptome alignments were then compared to ensure that only the best alignments were kept for each read. Based on the remaining alignments (genome or transcriptome, depending on requirements), expression estimates were computed (red boxes) either with the surveyed, model-based methods (3a), or count-based methods (RNA-seq: 3b, A-seq-2: 3c). Subsequently ('post-processing'), the raw numbers produced by the latter methods, as well as the true number of expressed transcripts in the synthetic dataset (as provided by Flux Simulator; gray arrow), were normalized, and the normalized expression estimates were extracted from the outputs of the surveyed model-based inference methods. Depending on the downstream analysis, expression estimates for transcripts and 3′ end processing sites ('Poly(A)') were aggregated and filtered (purple boxes; 4). To evaluate the performance of the surveyed methods (magenta boxes; 5), the accuracy of the surveyed transcripts abundance inference methods were analyzed by comparing the produced estimates to either the ground truth expression (synthetic data) or the A-seq-2-based estimates (experimental data). Additionally, runtime and memory consumption was evaluated. Steps at which either transcript/gene annotations (GENCODE) or transcript sequences (ENSEMBL) were used are marked with white triangles at the upper left corners. Refer to the Methods section and the main text for further details.



Supplementary Figure A.2: Multithreading efficiency and running time / memory footprint trade-off. Transcript isoform abundances were estimated with each of the indicated methods based on in silico-generated sequencing datasets. **(A)** The efficiency of multi-core use is indicated in terms of the speedup factor (ratio of running times when using 1 compared to 16 cores) for different sequencing depths. **(B and C)** Relationships between running time and memory footprint when processing 100 million reads with either 1 (B) or 16 (C) cores. Note that data for TIGAR2 are unavailable for (A) and (C), because the method does not support the use of multiple cores.



Supplementary Figure A.3: Accuracy of transcript isoform abundance estimates inferred from in silico-generated sequencing data. For each method, correlations between true and inferred transcript abundances are shown as heat density plots. The corresponding Spearman correlation coefficients (rs) are indicated. Estimates were produced based on the 30 million read dataset. **(A)** BitSeq. **(B)** CEM. **(C)** Cufflinks. **(D)** eXpress. **(E)** IsoEM. **(F)** MMSEQ. **(G)** RSEM. **(H)** rSeq. **(I)** Sailfish. **(J)** Scripture. **(K)** TIGAR2.

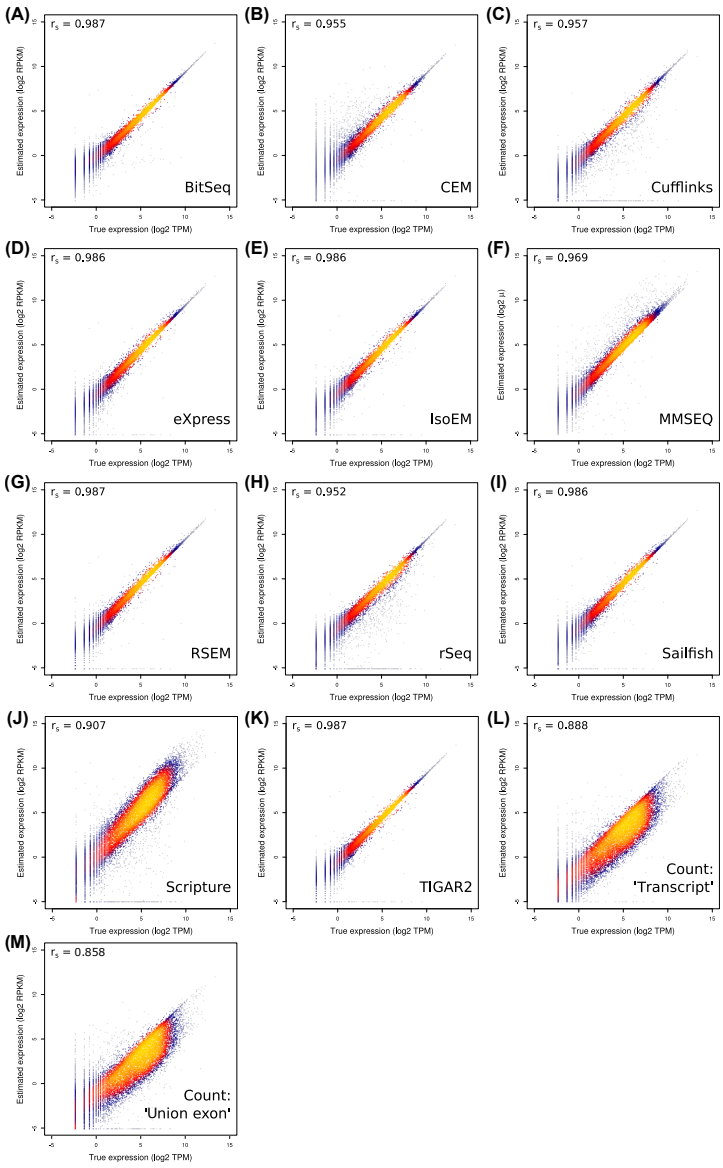
(A)

	Spearman					Pearson					RMSE				
	1 million reads	3 million reads	10 million reads	30 million reads	100 million reads	1 million reads	3 million reads	10 million reads	30 million reads	100 million reads	1 million reads	3 million reads	10 million reads	30 million reads	100 million reads
BitSeq	0.89	0.94	0.97	0.97	0.97	0.89	0.94	0.96	0.97	0.96	1.56	1.23	1.13	1.22	1.39
CEM	0.89	0.92	0.94	0.94	0.94	0.85	0.88	0.91	0.92	0.92	3.09	2.41	2.00	1.86	1.81
Cufflinks	0.76	0.77	0.77	0.77	0.77	0.75	0.76	0.76	0.76	0.76	4.09	3.60	3.33	3.23	3.21
eXpress	0.90	0.93	0.95	0.96	0.97	0.87	0.90	0.93	0.94	0.95	3.03	2.41	1.94	1.69	1.56
IsoEM	0.90	0.94	0.96	0.96	0.97	0.86	0.90	0.93	0.94	0.94	3.15	2.36	1.86	1.63	1.56
MMSEQ	0.90	0.92	0.93	0.93	0.93	0.85	0.89	0.90	0.90	0.90	3.75	2.91	2.31	2.03	1.87
RSEM	0.90	0.93	0.95	0.96	0.96	0.86	0.90	0.92	0.93	0.94	3.19	2.42	1.92	1.69	1.63
rSeq	0.87	0.90	0.92	0.92	0.92	0.83	0.87	0.89	0.90	0.90	3.52	2.80	2.37	2.18	2.12
Sailfish	0.89	0.93	0.95	0.96	0.96	0.86	0.89	0.92	0.93	0.94	3.22	2.50	1.99	1.79	1.67
Scripture	0.75	0.76	0.77	0.77	0.77	0.71	0.73	0.74	0.75	0.75	3.03	2.64	2.44	2.37	2.36
TIGAR2	0.90	0.93	0.95	0.96	0.96	0.86	0.90	0.92	0.94	0.94	3.27	2.50	1.99	1.74	1.65

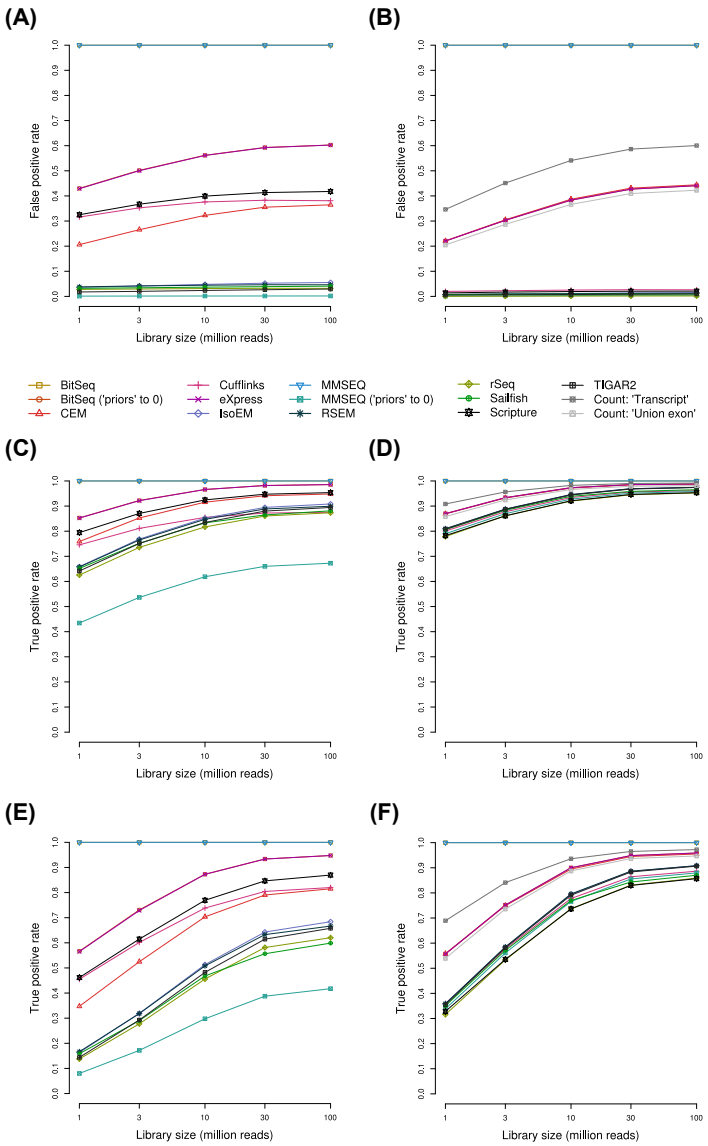
(B)

	Spearman					Pearson					RMSE				
	1 million reads	3 million reads	10 million reads	30 million reads	100 million reads	1 million reads	3 million reads	10 million reads	30 million reads	100 million reads	1 million reads	3 million reads	10 million reads	30 million reads	100 million reads
BitSeq	0.90	0.95	0.98	0.99	0.99	0.86	0.93	0.97	0.98	0.98	1.92	1.36	0.94	0.81	0.86
CEM	0.93	0.95	0.95	0.95	0.96	0.87	0.90	0.92	0.93	0.93	2.10	1.63	1.37	1.31	1.29
Cufflinks	0.94	0.95	0.96	0.96	0.96	0.89	0.91	0.93	0.93	0.93	2.46	1.93	1.62	1.53	1.53
eXpress	0.96	0.97	0.98	0.99	0.99	0.91	0.94	0.96	0.97	0.97	2.29	1.71	1.31	1.13	1.03
IsoEM	0.96	0.98	0.98	0.99	0.99	0.91	0.94	0.96	0.97	0.97	2.18	1.58	1.16	1.02	0.97
MMSEQ	0.95	0.97	0.97	0.97	0.97	0.89	0.93	0.95	0.95	0.95	1.83	1.44	1.19	1.13	1.13
RSEM	0.96	0.98	0.98	0.99	0.99	0.91	0.94	0.96	0.97	0.97	2.18	1.58	1.16	1.02	0.98
rSeq	0.93	0.94	0.95	0.95	0.95	0.88	0.90	0.92	0.93	0.93	2.63	2.08	1.78	1.68	1.65
Sailfish	0.96	0.98	0.98	0.99	0.99	0.91	0.94	0.96	0.97	0.97	2.23	1.62	1.22	1.08	1.02
Scripture	0.89	0.90	0.91	0.91	0.91	0.86	0.87	0.88	0.89	0.89	2.80	2.31	2.01	1.91	1.88
TIGAR2	0.96	0.98	0.98	0.99	0.99	0.91	0.94	0.96	0.97	0.97	2.21	1.61	1.22	1.09	1.03
Count: 'Transcript'	0.87	0.88	0.89	0.89	0.89	0.86	0.88	0.88	0.89	0.89	2.95	2.69	2.57	2.54	2.53
Count: 'Union exon'	0.84	0.85	0.86	0.86	0.86	0.82	0.84	0.85	0.85	0.85	3.03	2.76	2.64	2.62	2.61

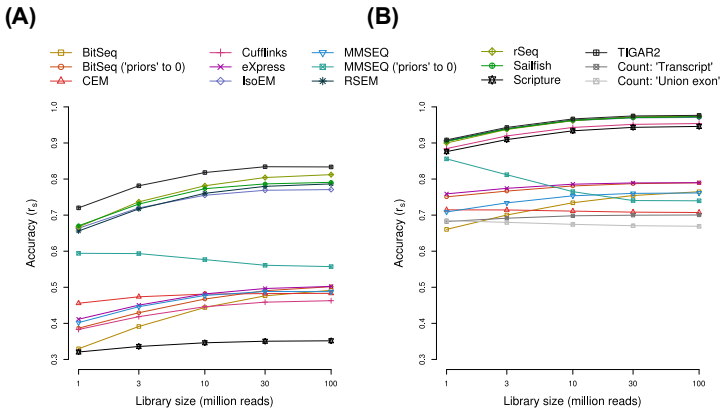
Supplementary Figure A.4: Comparison of different metrics for quantifying the accuracy of isoform abundance estimates. The accuracy of expression level estimates with respect to the ground truth was assessed by the Spearman and Pearson correlation coefficients, as well as the root mean square error (RMSE). The values obtained for expressed transcripts **(A)** and expressed genes **(B)** are plotted. Color intensities have been computed per column by scaling raw values such that the best value (high for correlation coefficients, low for RMSE) corresponds to the most intense and the worst to the least intense color.



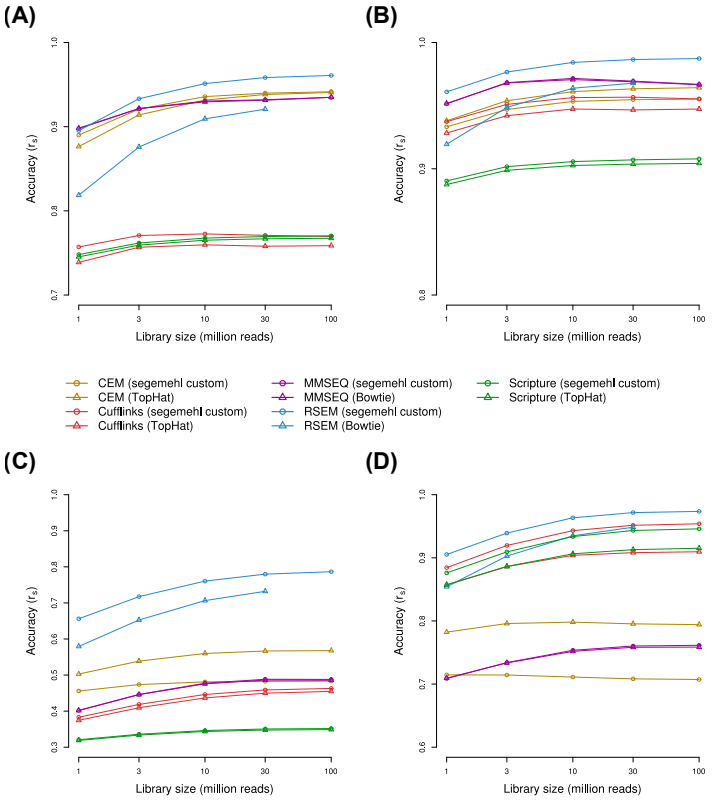
Supplementary Figure A.5: Accuracy of gene expression estimates inferred from in silico-generated sequencing data. As in Supplementary Figure A.3, but estimates were produced for genes instead of transcripts. **(A)** BitSeq. **(B)** CEM. **(C)** Cufflinks. **(D)** eXpress. **(E)** IsoEM. **(F)** MMSEQ. **(G)** RSEM. **(H)** rSeq. **(I)** Sailfish. **(J)** Scripture. **(K)** TIGAR2. **(L)** Counting method 'transcript'. **(M)** Counting method 'union exon'.



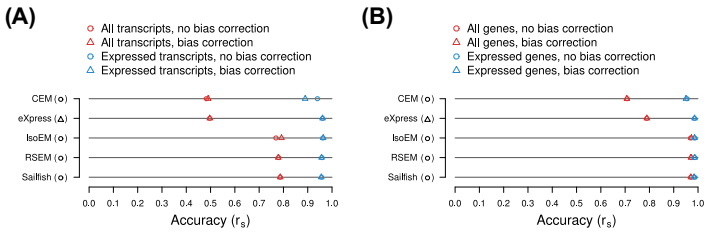
Supplementary Figure A.6: Accuracy of 'present calls'. The ability of each method to accurately determine whether a given transcript or gene is expressed was determined by calculating false positive **(A and B)** and true positive **(C through F)** rates across different sequencing depths. A transcript (A, C, and E) or gene (B, D, and F) was considered expressed, if it has - according to the ground truth - a non-zero expression. In contrast to A through D, where all features are considered, panels E and F show the true positive rates only for lowly expressed transcripts and genes (log2 TPM <0 and <1.1, respectively; compare expression bins in Figure 2.2). Note that by default, BitSeq and MMSEQ report small non-zero 'priors'. For these methods, we included modified estimates ('priors' to 0), in which a portion of these small values were set to zero according to simple algorithms (refer to the main text and the Methods section for details).



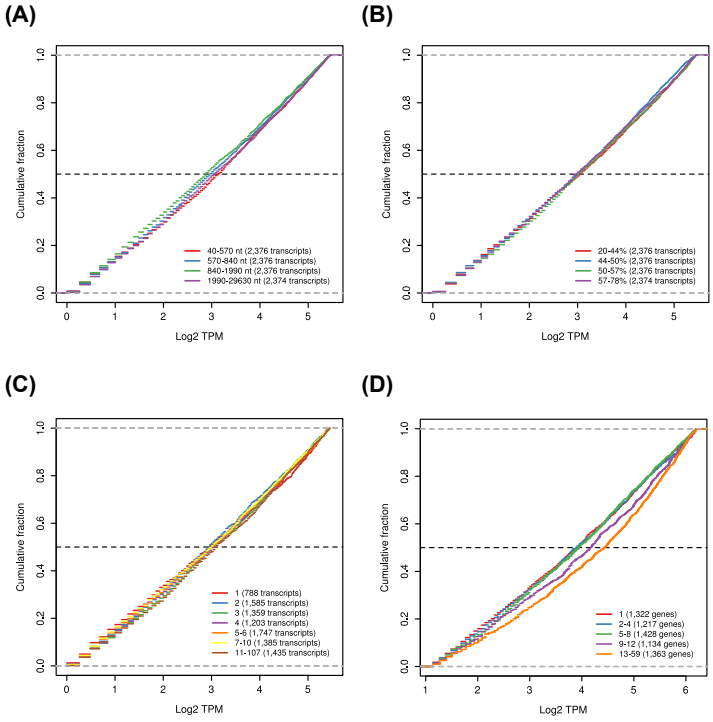
Supplementary Figure A.7: Accuracy of expression estimates across all transcripts and genes. As in Figure 2.2 A and B, but including, respectively, transcripts **(A)** and genes **(B)** that are not expressed according to the ground truth.



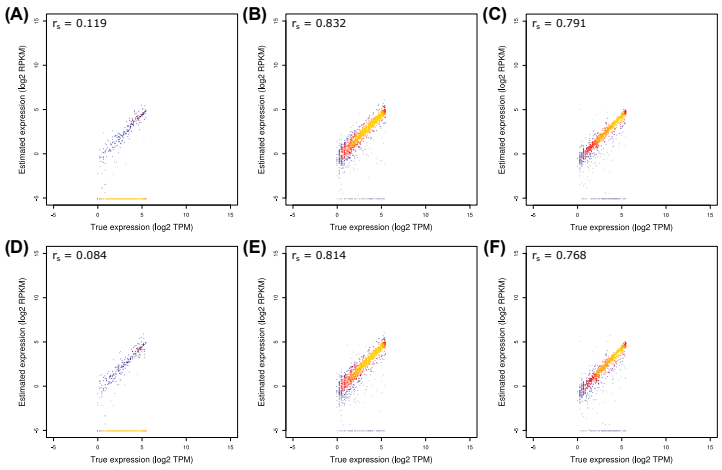
Supplementary Figure A.8: Effect of 'native' short-read aligners. For methods strongly recommending the use of a specific short-read aligner (CEM, Cufflinks, MMSEQ, Scripture) or using such an aligner internally by default (RSEM), expression levels inferred based on alignments obtained with the respective aligners were compared to the estimates produced following our own processing and alignment pipeline. Accuracies were calculated across different read depths as in Figure 2.2, either for expressed transcripts **(A)** or genes **(B)**, or for all transcripts **(C)** or genes **(D)**.



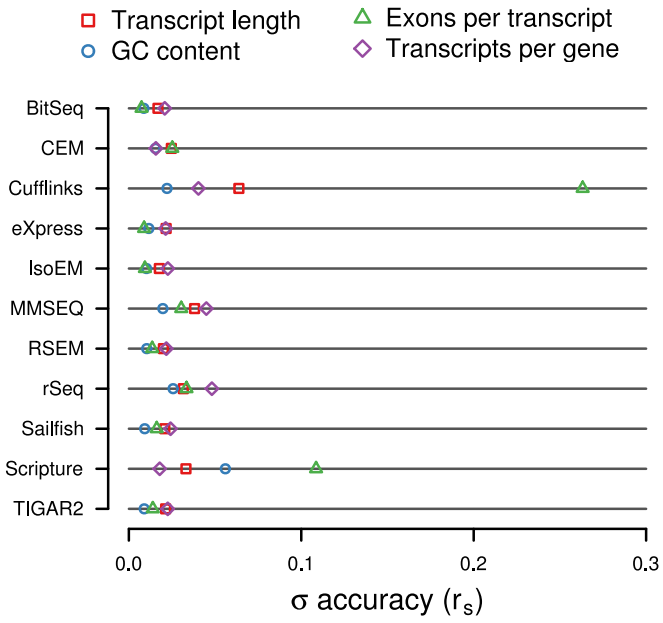
Supplementary Figure A.9: Impact of bias correction settings on simulated data. For methods where an optional sequencing/positional bias correction setting is implemented, we have compared estimation accuracies obtained when executing the programs with the respective options set or unset. Accuracies were calculated for 30 million reads as in Figure 2.2, either for transcripts **(A)** or genes **(B)**. Default settings (that were also used throughout this study if not indicated otherwise) are indicated in parentheses after the method name (circle: bias correction off, triangle: bias correction on). Note that Cufflinks also has a bias correction option (–frag-bias-correct; default: off). However, in our hands the program crashed when this option was specified.



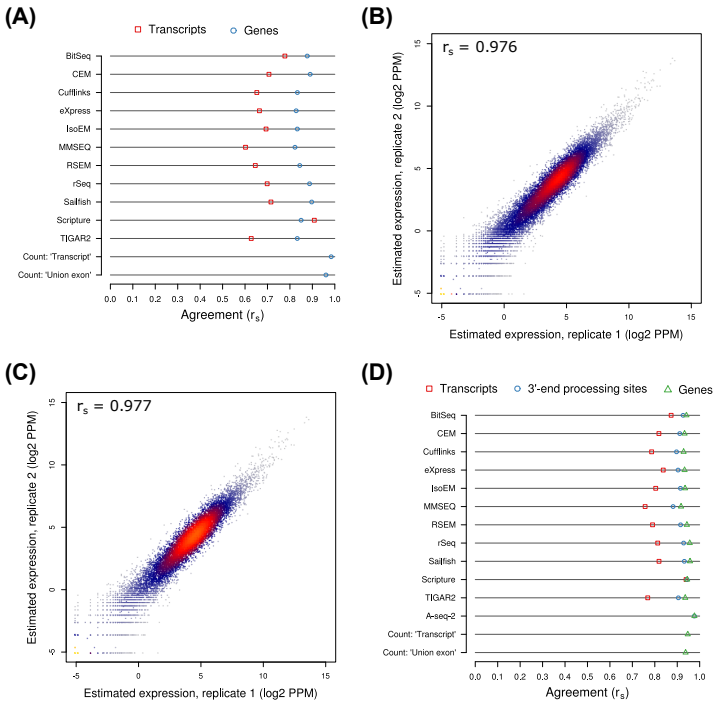
Supplementary Figure A.10: Expression level distributions across bins of transcripts and genes. All transcripts or genes expressed at levels of 0 < log2 TPM <5.5 were distributed across bins according to transcript length **(A)**, GC content **(B)**, the number exons per transcript **(C)**, and the number of transcripts per gene **(D)**. Ranges of the corresponding values covered by each bin are indicated in the legends to each chart, together with the number of features (transcripts or genes) they contain. The expression level distributions of the features in each bin are depicted as cumulative distribution functions.



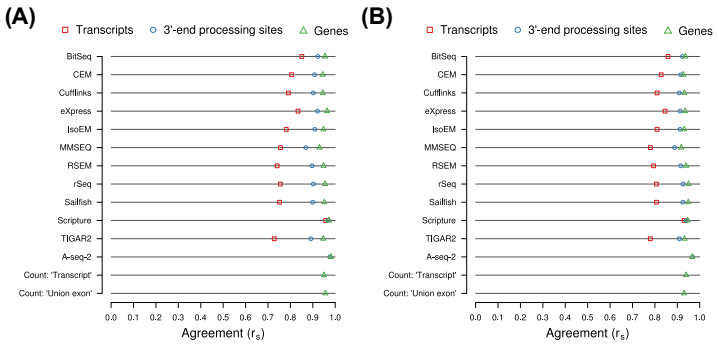
Supplementary Figure A.11: Cufflinks-based abundance estimates of single-exon transcripts. Cufflinks was used to infer transcript isoform expression levels from the alignments of 30 million in silico-generated reads. Alignments were produced either following our own segemehl-based pipeline **(A to C)** or by TopHat **(D to F)**. Estimated abundances are plotted against true abundances for transcripts expressed at 0 < log2 TPM <5.5 and comprising either one exon (A and D), two exons (B and E), or 11 or more exons (C and F). Heat map colors reflect the densities of data points and the corresponding Spearman correlation coefficients (rs) are indicated. For all single-exon transcripts expressed at 0 < log2 TPM <5.5, transcript isoform abundances as estimated by Cufflinks are plotted against true abundances.



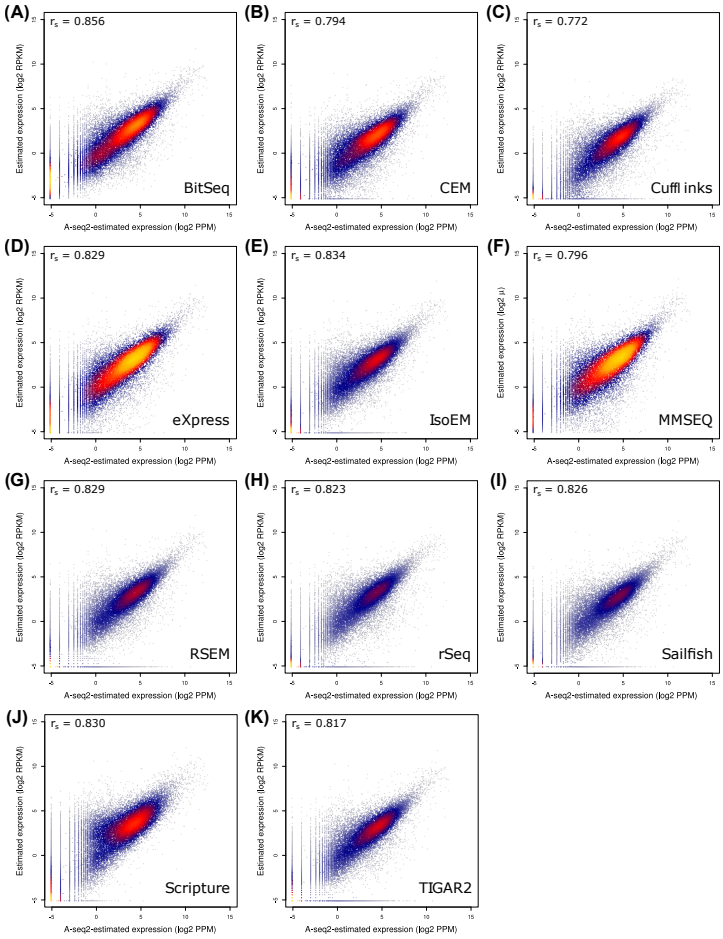
Supplementary Figure A.12: Impact of gene structural features on expression estimates. Transcripts and genes have been distributed over different bins according to the indicated structural features (see Figure 2.3 and main text). The variation between estimation accuracies for these bins are indicated in terms of the standard deviations σ of the Spearman correlation coefficients between ground truth and estimates.



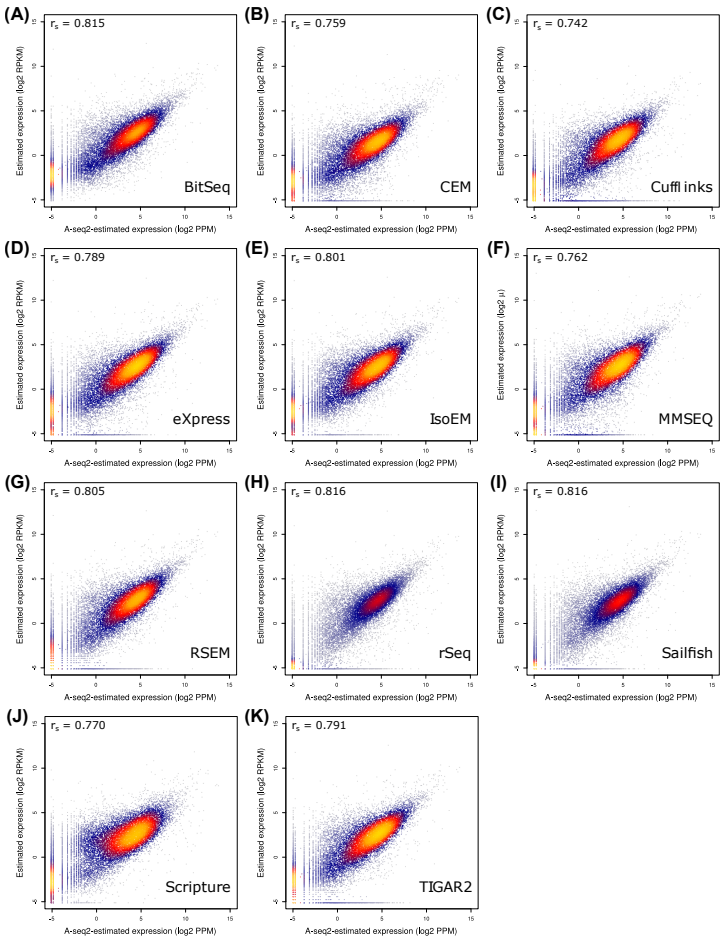
Supplementary Figure A.13: Agreement between expression level estimates for replicates of NIH/3T3 cells. Transcript isoform and gene abundances were estimated with each of the indicated methods based on RNA-seq data obtained from two biological replicates of murine NIH/3T3 cells. **(A)** The agreement between expression estimates for the two replicates are indicated as Spearman correlation coefficients rs, both at the level of transcripts and genes. **(B)** A-seq-2-based 3′ end processing site expression level estimates for the two replicates are plotted against each other. The Spearman correlation coefficient rs is indicated. **(C)** As in (B), but estimates are compared at the level of gene expression. **(D)** As in (A), but with the addition of 3′ end processing site abundances. For computing expression estimates for either feature type (transcript, 3′ end processing site, and gene), only those transcripts are considered that end in annotated 3′ end processing sites (see main text and Methods for details).



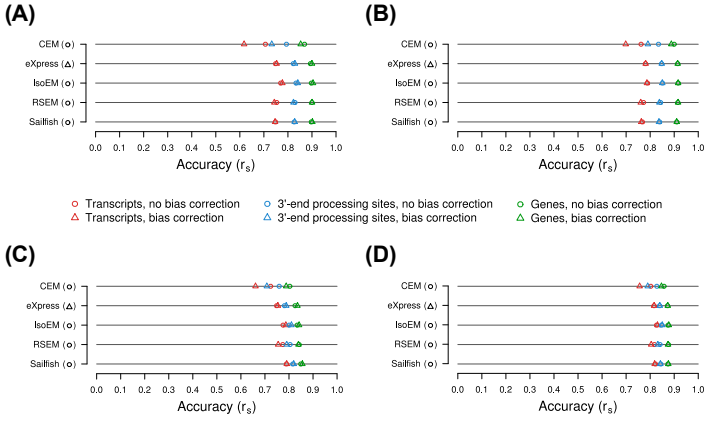
Supplementary Figure A.14: Replicate agreement between abundance estimates for features corresponding to expressed 3′ end processing sites. As in Figure 2.4 D and Supplementary Figure A.13 D, but with the further requirement that the considered transcripts need to end in annotated 3′ end processing sites that show evidence of expression, according to the A-seq-2 analysis. Results are shown for replicates of **(A)** human Jurkat cells and **(B)** murine NIH/3T3 cells.



Supplementary Figure A.15: Accuracy of 3′ end processing site abundance estimates inferred from Jurkat sequencing data. Transcript abundances inferred by the surveyed methods from RNA-seq libraries prepared from human Jurkat cells (replicate 1) were aggregated by 3′ end processing sites and plotted against the corresponding estimates obtained by the analysis of A-seq-2 sequencing data. Heat map colors represent data point densities and Spearman correlation coefficients (rs) are indicated. **(A)** BitSeq. **(B)** CEM. **(C)** Cufflinks. **(D)** eXpress. **(E)** IsoEM. **(F)** MMSEQ. **(G)** RSEM. **(H)** rSeq. **(I)** Sailfish. **(J)** Scripture. **(K)** TIGAR2.



Supplementary Figure A.16: Accuracy of 3′ end processing site abundance estimates inferred from NIH/3T3 sequencing data. As in Supplementary Figure A.15, but data were from murine NIH/3T3 cells.

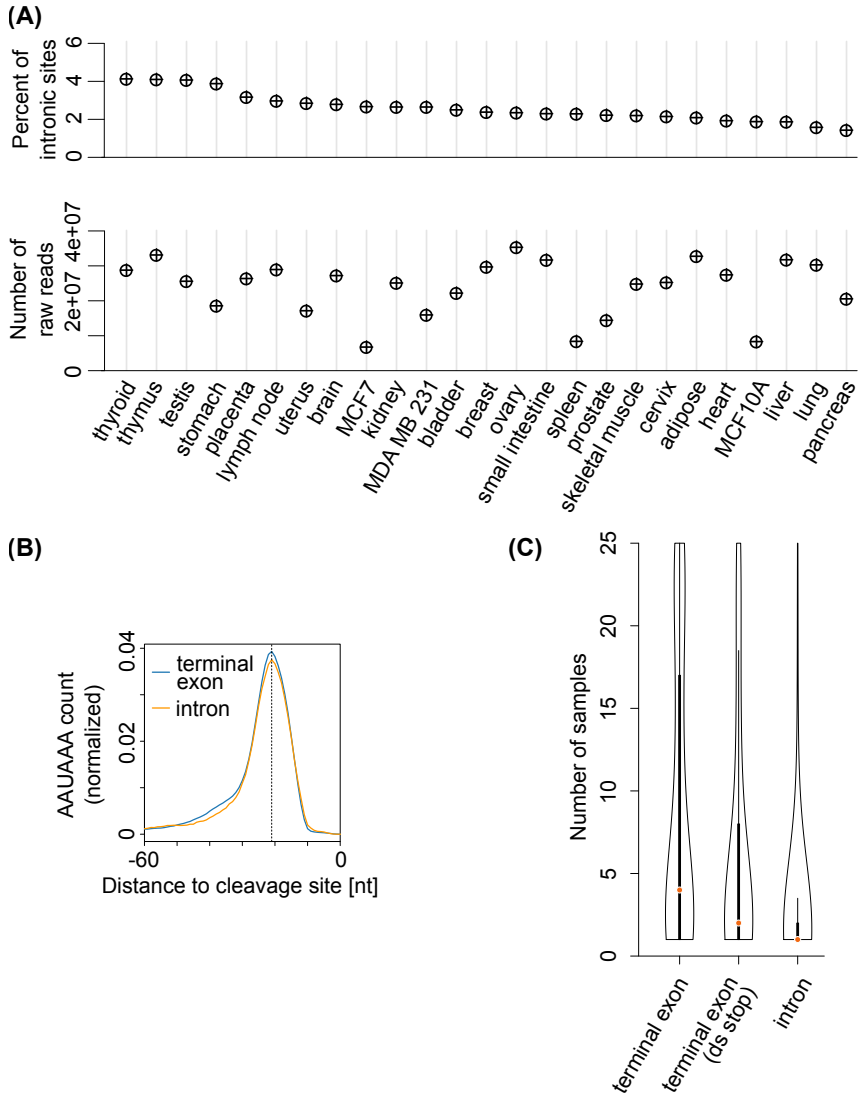


Supplementary Figure A.17: Impact of bias correction settings on abundance estimates from experimental data. As in Supplementary Figure A.9, but expression estimates were obtained for human (A and B) or mouse (C and D) cells and also include estimation accuracies on the level of 3′ end processing sites. Spearman correlation coefficients were calculated by comparison to A-seq-2 estimates (see Figure 2.5) rather than the simulation ground truth. (A) Jurkat data, replicate 1. (B) Jurkat data, replicate 2. (C) NIH/3T3 data, replicate 1. (D) NIH/3T3 data, replicate 2.

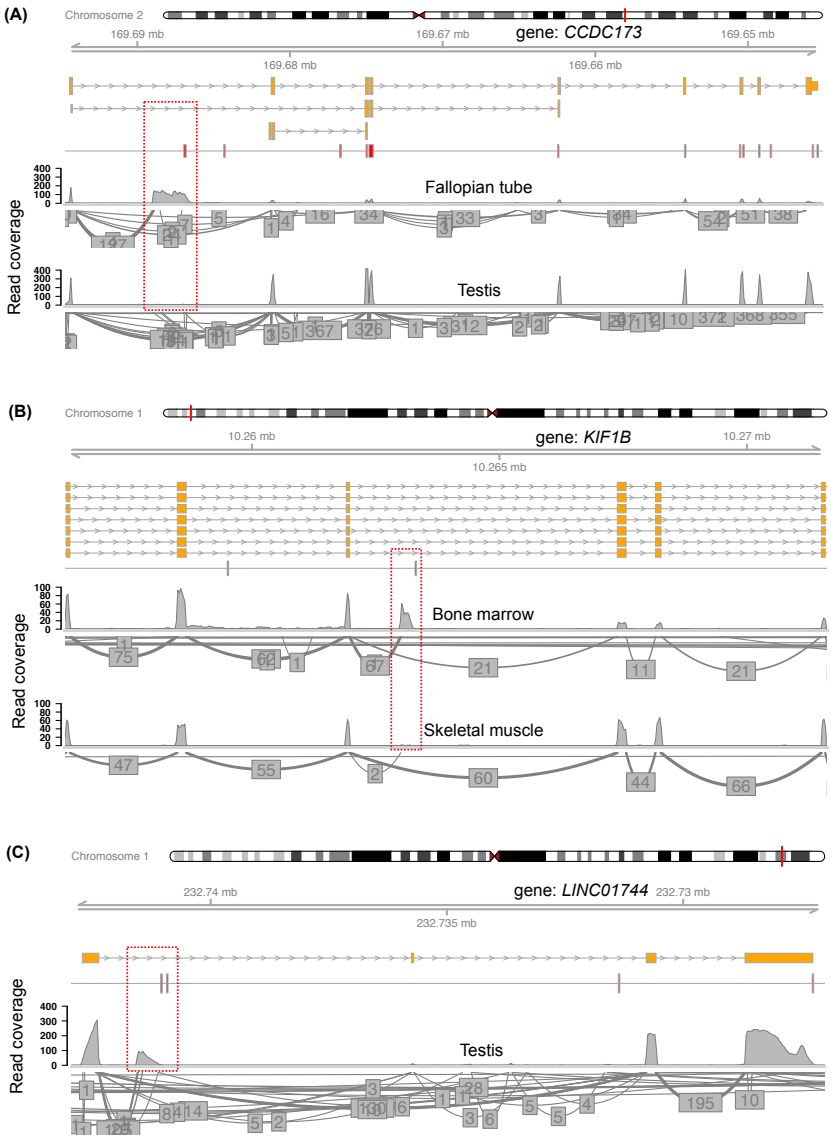
B

TECTOOL SUPPLEMENTS

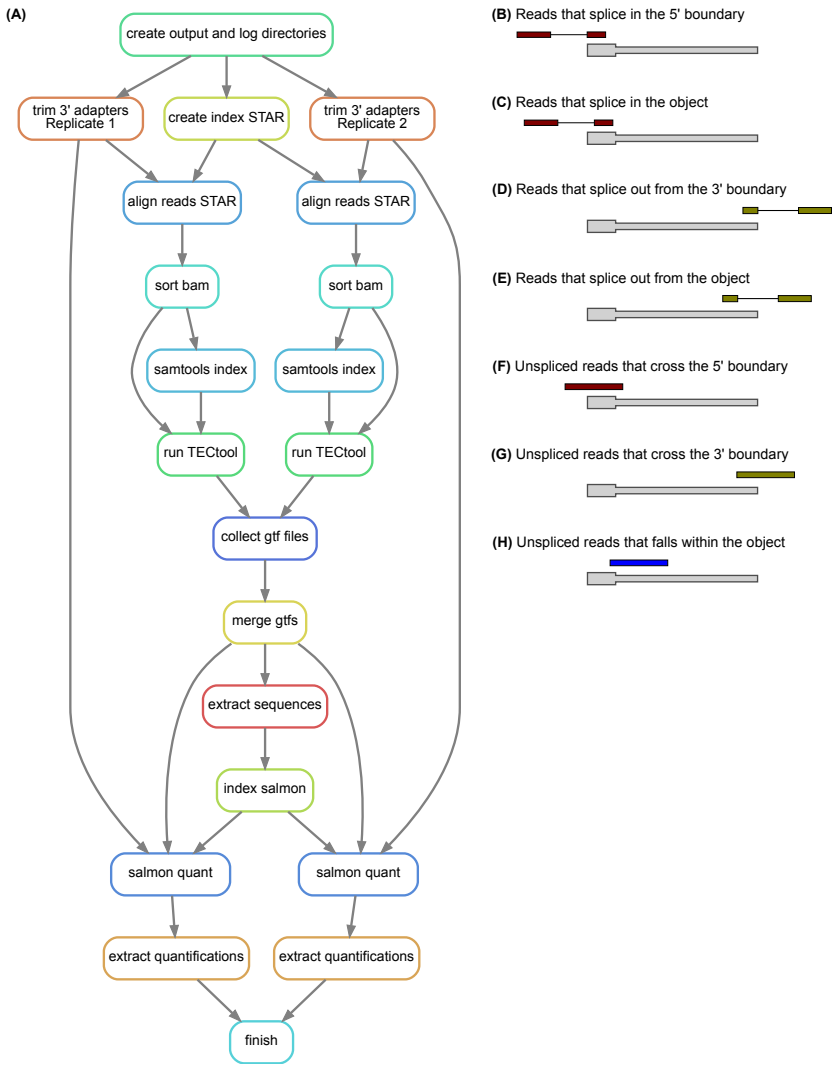
Supplementary material to chapter 3.



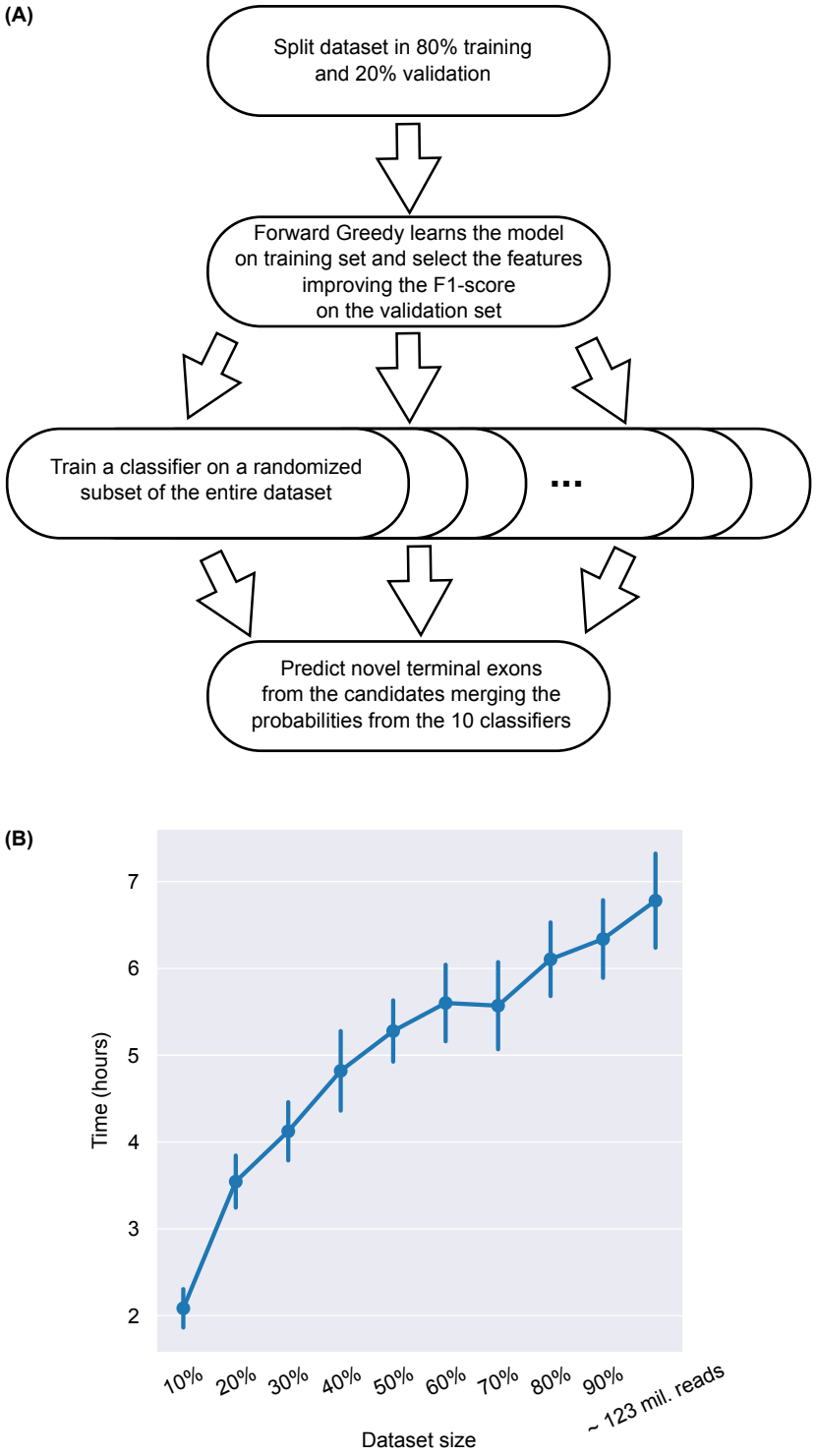
Supplementary Figure B.1: 'Intronic' poly(A) sites are processed in a tissue-specific manner. (A) Top panel: Percentage of 'intronic' PAS in individual samples in the data sets obtained with the SAPAS protocol [193]. Bottom panel: corresponding sequencing depths. **(B)** Position-dependent frequency of the canonical poly(A) signal ('AAUAAA') upstream of the 'intronic' poly(A) sites (orange) and of poly(A) sites corresponding to annotated terminal exons (blue) from the study introduced in (A). The usual position of the poly(A) signal at 21 nts upstream of the cleavage site is indicated by the dashed, vertical line. **(C)** Distribution of the number of distinct samples (from panel B) in which individual PAS were observed, for different types of PAS; 'introns' - PAS from genomic regions currently annotated as intronic; 'terminal exon (ds stop)' - PAS from annotated terminal exons that are located upstream of an annotated stop codon in the corresponding gene; 'terminal exon' - PAS from terminal exons with no stop codon annotated downstream.



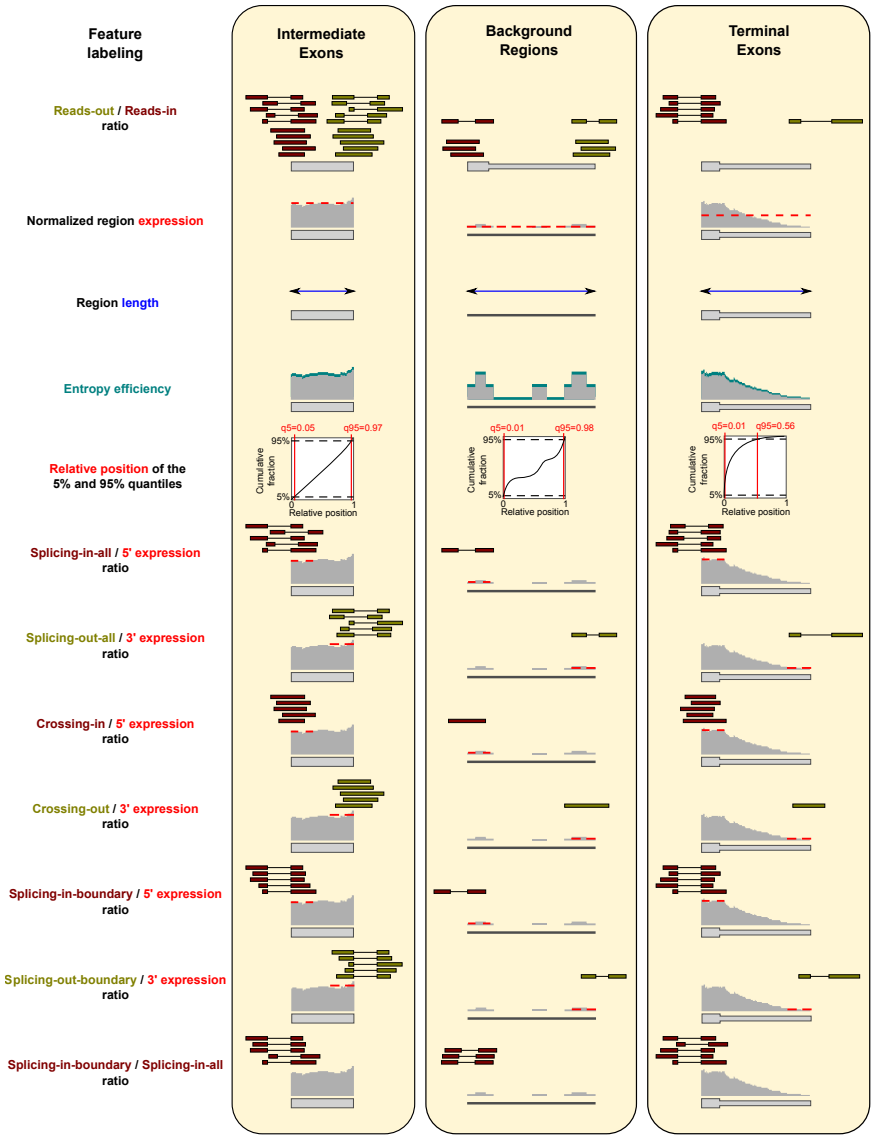
Supplementary Figure B.2: Sashimi plots of gene structures inferred from the RNA-seq data from different tissues. (A) The Coiled-coil Domain Containing 173 (CCDC173) gene locus with the annotated ENSEMBL transcript (orange), PAS from the PolyAsite atlas (red lines), and densities of mRNA reads (gray) from fallopian tube and testis samples. Gray arcs indicate spliced reads with their corresponding numbers. The novel terminal exon (red dotted box) is expressed in the fallopian tube, but not in testis, indicating a sex-dependent isoform switch. Note: Same as Figure 3.2A, but showing all spliced reads. **(B)** Same representation for part of the Kinesin Family Member 1B (*KIF1B*) gene locus. The novel terminal exon (red dotted box) is mainly expressed in bone marrow. **(C)** Similar for the locus of lincRNA *LINC01744*.



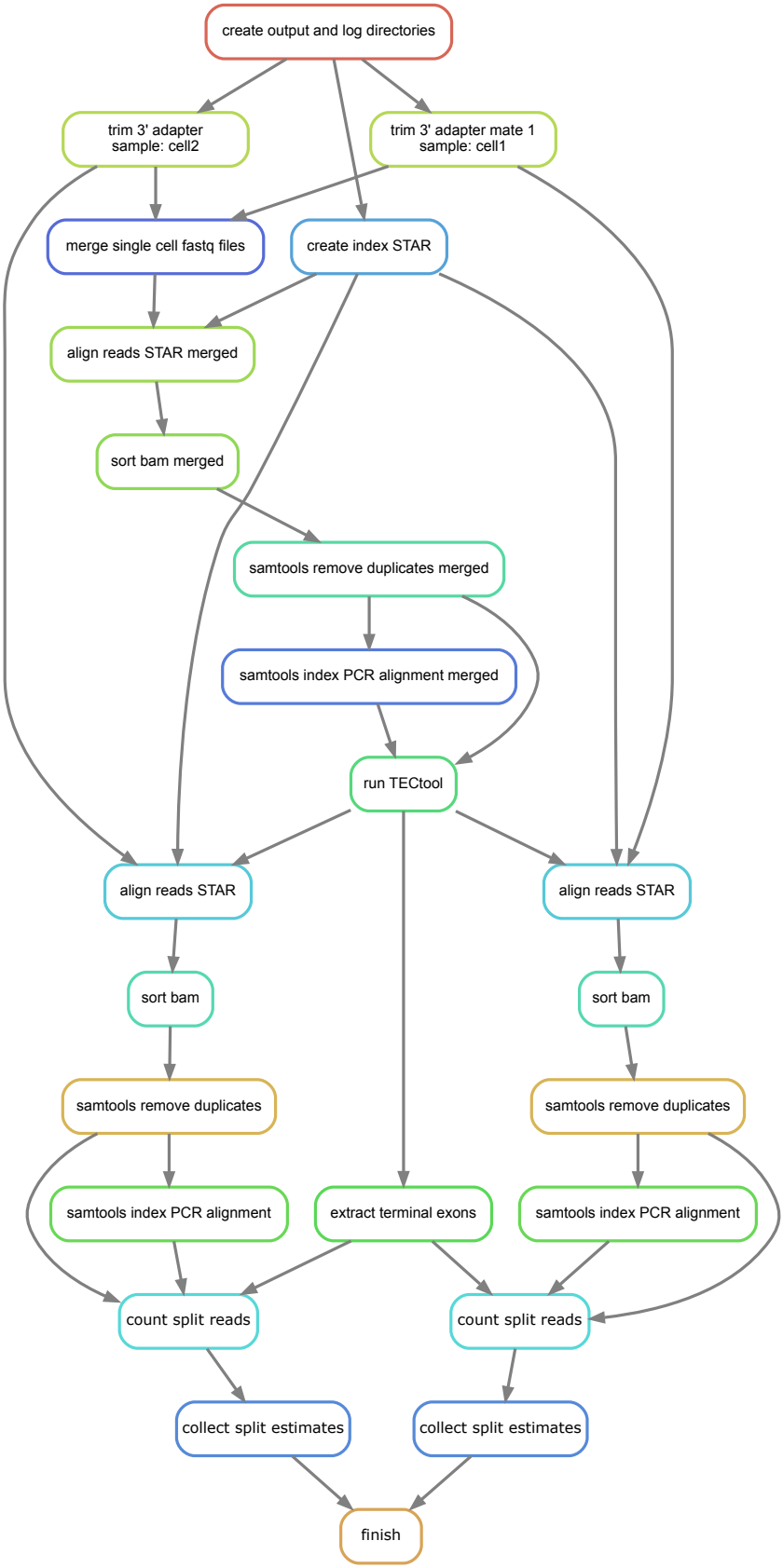
Supplementary Figure B.3: TECtool analysis for bulk single-end or paired-end RNA-seq reads. (A) Graphic representation of the analysis flow for two replicates. **(B-H)** Features calculated by TECtool for annotated and putative terminal exons. The region from which these statistics are calculated is referred to as 'object'. **(B) Splicing-in-boundary:** Reads that splice from an upstream region to the 5′ boundary of the object. **(C) Splicing-in-all:** Reads that splice from an upstream region anywhere within the object. **(D) Splicing-out-boundary:** Reads that splice from the 3′ boundary of the object to a downstream region. **(E) Splicing-out-all:** Reads that splice out from anywhere within the object to a downstream region. **(F) Crossing-in-boundary:** Unspliced reads that overlap the 5′ boundary of the object. **(G) Crossing-out-boundary:** Unspliced reads that overlap the 3′ boundary of the object. **(H) Unspliced-within-boundaries:** Unspliced reads that are contained in the object.



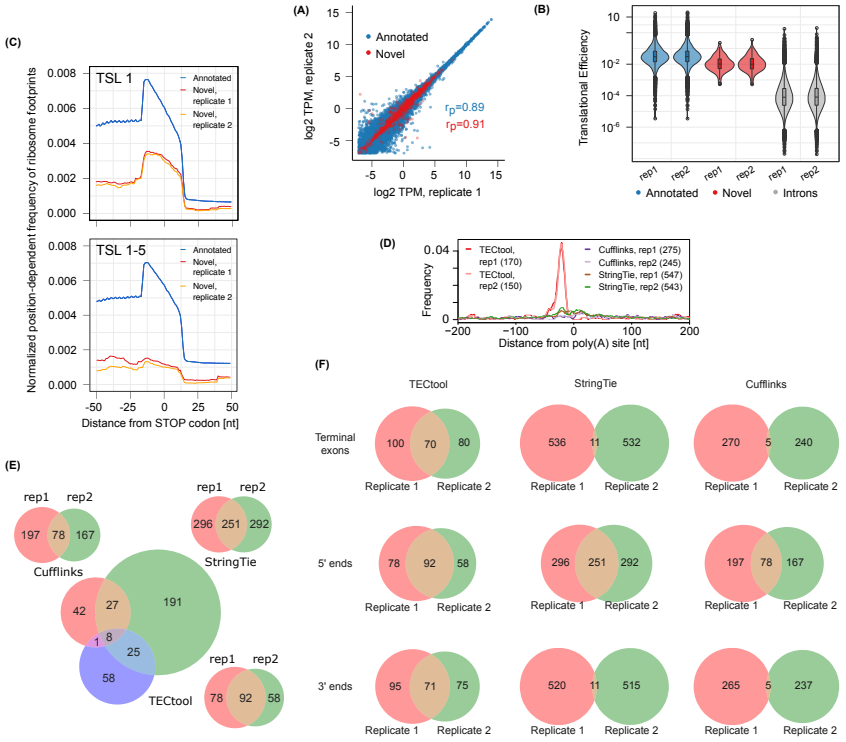
Supplementary Figure B.4: Overview of the region classification model in TECtool. (A) Flow chart of the machine learning algorithm in TECtool. **(B)** Analysis of TECtool running time. A data set of approximately 123 million reads was subsampled in increments of 10% (starting from approximately 12 million reads) and analyzed running TECtool on a single CPU. The analysis was repeated 10 times for each data set size. Shown are the mean and standard deviation over the 10 runs for each data set size.



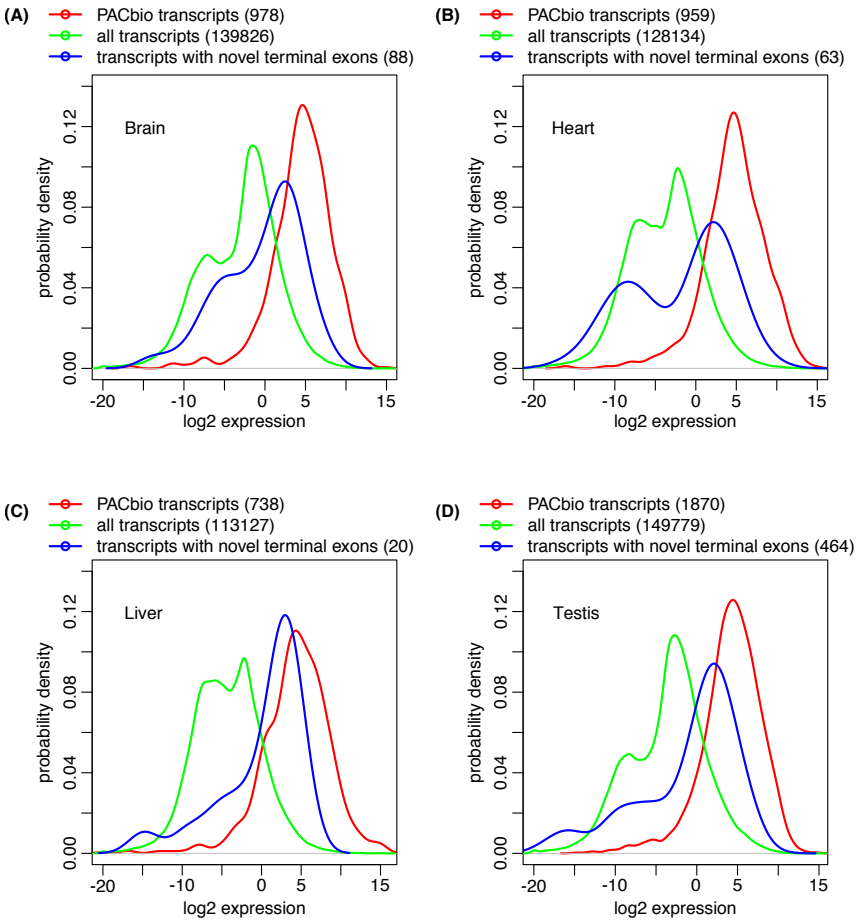
Supplementary Figure B.5: Features used in the model. Schematic representation of the features that are used to construct the model and then classify regions into terminal exons, intermediate exons or background regions.



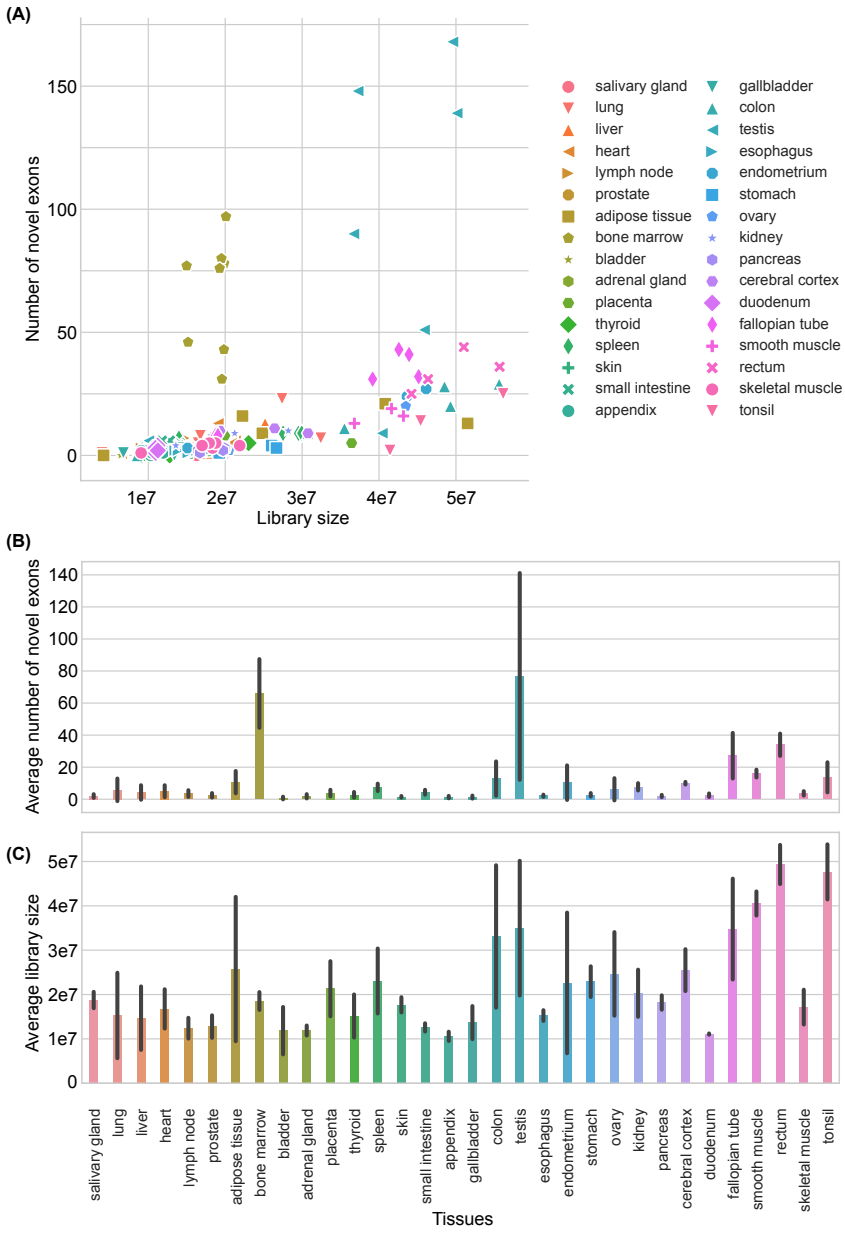
Supplementary Figure B.6: TECtool analysis flow for single cell data. Example of TECtool analysis of two individual cells.



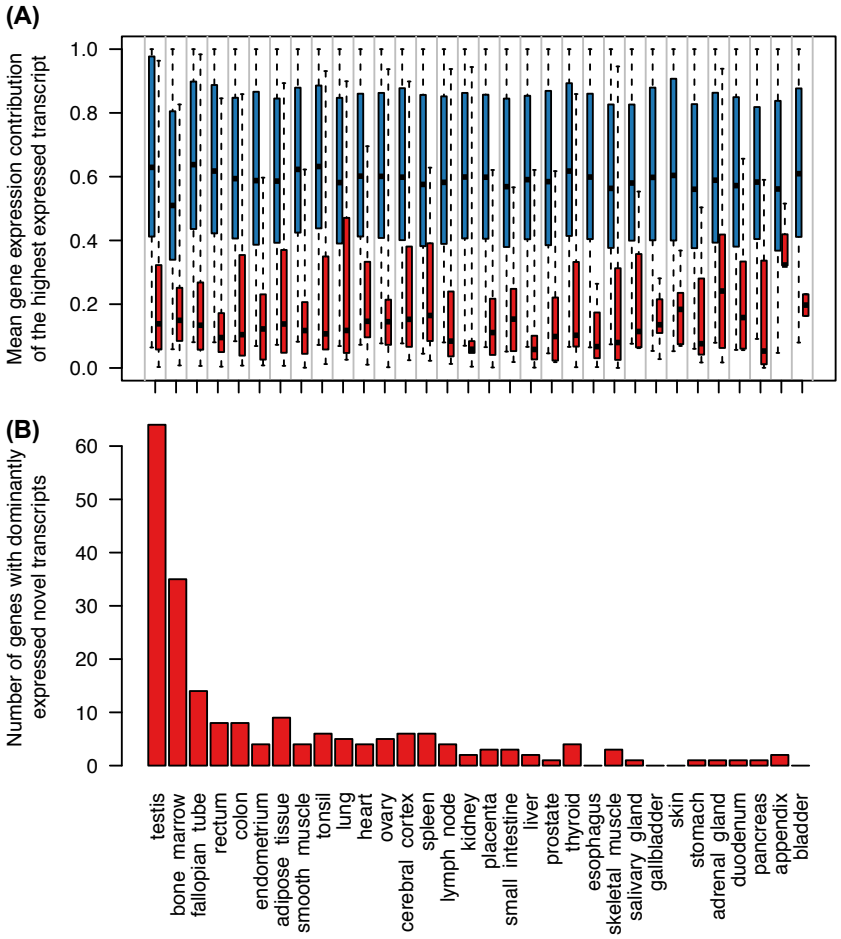
Supplementary Figure B.7: Evaluation of TECtool performance. (A) Scatter plot of estimated expression levels of already annotated transcripts (ENSEMBL v87, transcript support level 1 (TSL1), blue, 41,676 transcripts) and of transcripts ending at TECtool-identified terminal exons (red, 893 transcripts), in biological replicates of RNA-seq from HEK 293 cells (r_P indicate the corresponding Pearson correlations). **(B)** Translational efficiencies computed for annotated terminal exons, novel terminal exons and intronic regions (two-tailed t-test p-values for pairwise comparisons of regions based on TSL1, novel versus intron replicate 1 (rep1): 5.6e-85; replicate 2 (rep2): 2.3e-84, and annotated versus novel rep1: 1.7e-20; rep2: 2.2e-20). Boxes indicate the interquartile range (IQR) with the line corresponding to the median, whiskers correspond to the most extreme value that is within 1.5 times the IQR from the hinge and outliers beyond this range are shown as individual points. **(C)** Normalized position-dependent frequencies of ribosome footprints around STOP codons of annotated (upper panel TSL1, lower panel TSL1-5) or novel transcripts. **(D)** Smoothened (+/- 5 nucleotides) frequency profiles of the canonical poly(A) signal ('AAUAAA') around 3' ends of transcripts predicted as novel relative to TSL1-5 by TECtool, StringTie and Cufflinks, respectively. **(E)** Venn diagrams showing the number of unique terminal exons defined only by their 5' ends, that were predicted by Cufflinks, StringTie and TECtool from the two replicate HEK 293 RNA-seq data sets using TSL1-5 annotation. The three-circle Venn diagram shows the relationship between 5' end-defined terminal exons that were predicted in both replicates by the above mentioned tools. **(F)** Venn diagrams reflecting the reproducibility of terminal exon prediction by TECtool, StringTie and Cufflinks (using again ENSEMBL v87 TSL1-5 annotation). Two independent biological replicates were analyzed with the mentioned tools to identify novel terminal exons. The overlap was then determined when exons were uniquely defined based on both their 5' and 3' genome coordinate, or based only on the 5' end, or only on the 3' end.



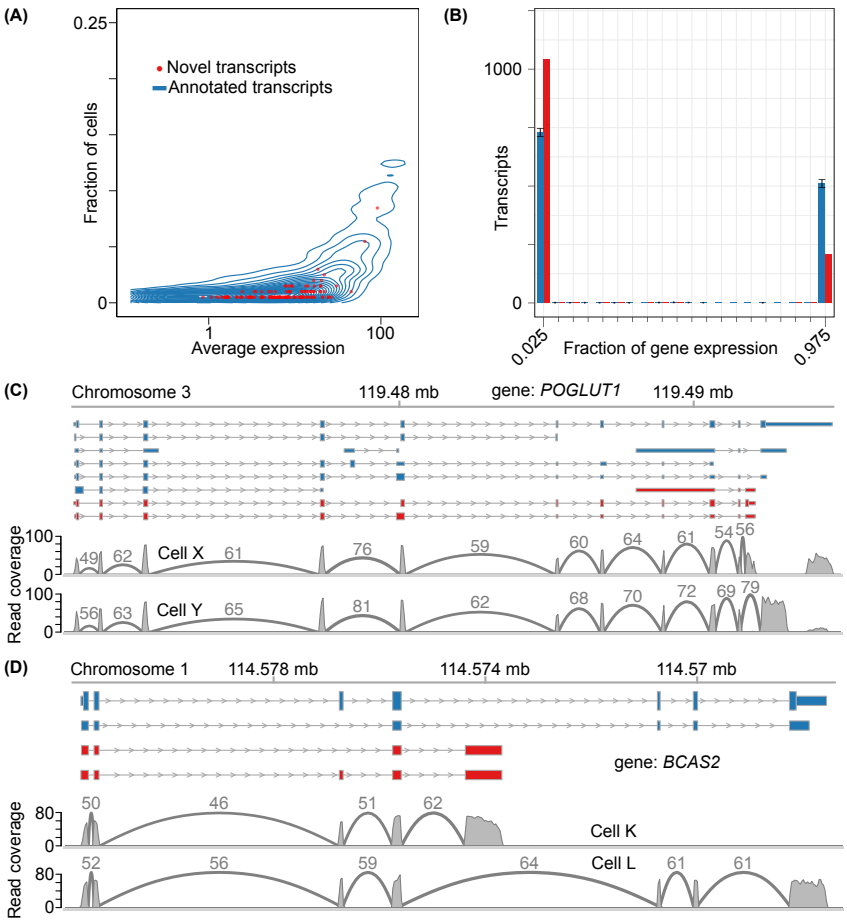
Supplementary Figure B.8: Distribution of expression levels inferred by Salmon [199] from short read sequencing data. RNA-seq was carried out from **(A)** brain, **(B)** heart, **(C)** liver and **(D)** testis samples, and distributions are shown for various transcript sets: all annotated transcripts (green), transcripts sequenced on the PacBio platform from the corresponding samples (red), and transcripts with novel terminal exons predicted by TECtool (blue). The number of transcripts is indicated in parentheses.



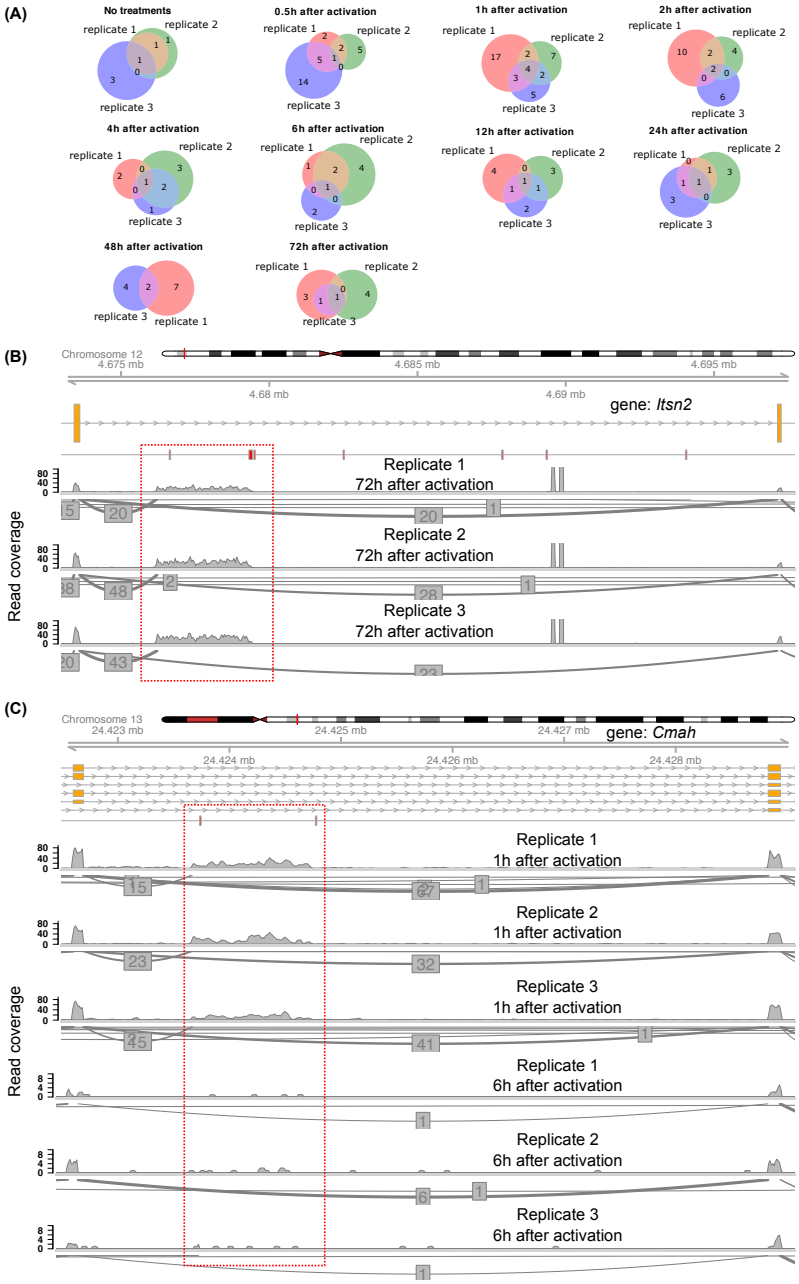
Supplementary Figure B.9: Summary of RNA-seq samples from the protein atlas data set. (A) Scatter plot of the number of mapped reads (mate 1) and number of novel exons identified by TECtool. Pearson's r = 0.56, p-value = 3.93e-17. Spearman's r = 0.73, p-value = 6.44e-34. **(B)** Barplots of the number of novel exons identified from individual tissues (black: error bars indicating standard deviation computed based on replicate samples) **(C)** and corresponding library sizes (black: error bars indicating standard deviation computed based on replicate samples). **Note for (A-C):** Number of samples used for each tissue indicated in parenthesis: salivary gland (6), lung (8), liver (5), heart (9), lymph node (13), prostate (7), adipose tissue (6), bone marrow (8), bladder (4), adrenal gland (6), placenta (7), thyroid (9), spleen (5), skin (6), small intestine (8), appendix (6), gallbladder (6), colon (8), testis (8), esophagus (6), endometrium (6), stomach (4), ovary (5), kidney (4), pancreas (4), cerebral cortex (3), duodenum (4), fallopian tube (6), smooth muscle (3), rectum (4), skeletal muscle (6), tonsil (3).



Supplementary Figure B.10: Update update update: Expression of TEC-tool identified transcripts across 32 human tissues. (A) Distribution of the mean gene expression contribution of the, on average, most highly expressed annotated (blue) or novel (red) isoform to the total expression of the corresponding gene in the indicated tissue. **(B)** Number of genes for which a novel transcript is, on average, the dominant expressed isoform. For (A) and (B) only novel transcripts having a median expression >1 TPM within a specific tissue were considered.



Supplementary Figure B.11: TECtool identifies novel isoforms that are expressed in subsets of single cells. (A) Fractions of cells expressing annotated (blue density) or novel (red dots) transcripts, as a function of the average expression of these transcripts across all cells. **(B)** Histograms of the number of transcripts which contribute a specific fraction of expression of their corresponding gene. Novel transcripts are shown in red, annotated transcripts in blue. We subsampled 20 times the set of annotated transcripts with a mean expression across cells similar to that of novel transcripts (subset size equal to the size of the novel transcripts set), and computed means and standard deviations over the 20 resamplings. Only reads that spliced into terminal exons were used to estimate the expression of transcripts containing the respective terminal exons. Furthermore, we only considered cases where there were at least two distinct reads that could be counted towards the expression of a given gene. **(C)** Sashimi plot of the locus of the O-glucosyltransferase 1 (*POGLUT1*) gene with the annotated ENSEMBL transcripts (blue), the novel transcripts predicted by TECtool (red), and RNA-seq read densities (gray) within two different cells. Gray arcs indicate spliced reads with their corresponding numbers. The first track indicates that a novel transcript is expressed in cell X, whereas the second track indicates that another cell, Y, expresses the annotated transcript. **(D)** Similar to (C) but for the Pre-mRNA Processing Factor (*BCAS2*) gene locus.



Supplementary Figure B.12: TECtool analysis of an RNA-seq data set obtained from mouse CD4+ T cells. (A) Overlap of novel terminal exons sets identified by TECtool from 3 replicate samples for each of the following CD4+ populations: untreated, and the at different time points following activation: 0.5, 1, 2, 4, 6, 12, 24, 48 and 72 hours. From replicate 2 of CD4+ cells 48 hours following activation, no novel terminal exons were identified. **(B)** Sashimi plots of gene structures inferred from the mouse CD4+ cell sequencing data. Part of the Intersectin 2 (*Itsn2*) gene locus with the annotated ENSEMBL transcript (orange), PAS from the PolyAsite atlas (`http://www.polyasite.unibas.ch`, red tick marks on the track under the gene structure), and densities of mRNA reads (gray) from 3 replicates of CD4+ T cells, 72 hours after activation. Gray arcs indicate spliced reads with their corresponding numbers. Red dotted box shows the novel terminal exon. **(C)** Same representation as in (B) but for the Cytidine monophospho-N-acetylneuraminic acid hydroxylase (*Cmah*) gene locus.

C

PUBLICATIONS AND CONTRIBUTION

This PhD thesis is based on the following publications:

1. **Comparative assessment of methods for the computational inference of transcript isoform abundance from RNA-seq data**
   Alexander Kanitz*, Foivos Gypas*, Andreas J. Gruber, Andreas R. Gruber, Georges Martin and Mihaela Zavolan. Genome Biology. (2015) [200]
   * equal contribution

2. **Terminal exon characterization with TECtool reveals an abundance of cell-specific isoforms**
   Andreas J. Gruber*, Foivos Gypas*, Andrea Riba, Ralf Schmidt, and Mihaela Zavolan. Nat. Methods. (2018) [207]
   * equal contribution

Both of the my projects were relatively large, and were developed together with colleagues from the group of Mihaela Zavolan. In both cases I was one of the two main authors, which is reflected in the shared first authorship. In the project described in chapter 2, I contributed to the design of the study, I generated the synthetic datasets, I communicated with the developers, I installed and executed some of the surveyed programs, analyzed resulting data, and I wrote sections of the manuscript. For the project described in chapter 3, I contributed to the design of the study, I co-developed the method, I packaged the tool, I automated the analysis flow, and carried out analyses to validate the method and demonstrate its performance. I further contributed to writing the manuscript.

BIBLIOGRAPHY

[1] E S Lander et al. "Initial sequencing and analysis of the human genome." en. In: *Nature* 409.6822 (Feb. 2001), pp. 860–921.

[2] J C Venter et al. "The sequence of the human genome." en. In: *Science* 291.5507 (Feb. 2001), pp. 1304–1351.

[3] Kim D Pruitt, Tatiana Tatusova, and Donna R Maglott. "NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins." en. In: *Nucleic Acids Res.* 35.Database issue (Jan. 2007), pp. D61–5.

[4] Rasmus Nielsen, Joshua S Paul, Anders Albrechtsen, and Yun S Song. "Genotype and SNP calling from next-generation sequencing data." In: *Nat. Rev. Genet.* 12.6 (2011), p. 443.

[5] Sara Goodwin, John D McPherson, and W Richard McCombie. "Coming of age: ten years of next-generation sequencing technologies." en. In: *Nat. Rev. Genet.* 17.6 (May 2016), pp. 333–351.

[6] Sara A Byron, Kendall R Van Keuren-Jensen, David M Engelthaler, John D Carpten, and David W Craig. "Translating RNA sequencing into clinical diagnostics: opportunities and challenges." en. In: *Nat. Rev. Genet.* 17.5 (May 2016), pp. 257–271.

[7] Zhong Wang, Mark Gerstein, and Michael Snyder. "RNA-Seq: a revolutionary tool for transcriptomics." en. In: *Nat. Rev. Genet.* 10.1 (Jan. 2009), pp. 57–63.

[8] Eesha Sharma, Tim Sterne-Weiler, Dave O'Hanlon, and Benjamin J Blencowe. "Global Mapping of Human RNA-RNA Interactions." en. In: *Mol. Cell* 62.4 (May 2016), pp. 618–626.

[9] Shailesh Kumar, Angie Duy Vo, Fujun Qin, and Hui Li. "Comparative assessment of methods for the fusion transcripts detection from RNA-Seq data." en. In: *Sci. Rep.* 6 (Feb. 2016), p. 21597.

[10] Sung Wook Chi, Julie B Zang, Aldo Mele, and Robert B Darnell. "Argonaute HITS-CLIP decodes microRNA-mRNA interaction maps." en. In: *Nature* 460.7254 (July 2009), pp. 479–486.

[11] Markus Hafner et al. "Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP." In: *Cell* 141.1 (Apr. 2010), pp. 129–141.

[12] Julian Konig, Kathi Zarnack, Gregor Rot, Tomaz Curk, Melis Kayikci, Blaz Zupan, Daniel J Turner, Nicholas M Luscombe, and Jernej Ule. "iCLIP–transcriptome-wide mapping of protein-RNA interactions with individual nucleotide resolution." en. In: *J. Vis. Exp.* 50 (Apr. 2011).

[13] Toshiyuki Shiraki et al. “Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage.” In: *Proc. Natl. Acad. Sci. U. S. A.* 100.26 (Dec. 2003), pp. 15776–15781.

[14] Peter J Shepard, Eun-A Choi, Jente Lu, Lisa A Flanagan, Klemens J Hertel, and Yongsheng Shi. “Complex and dynamic landscape of RNA polyadenylation revealed by PAS-Seq.” In: *RNA* 17.4 (Apr. 2011), pp. 761–772.

[15] Adnan Derti, Philip Garrett-Engele, Kenzie D Macisaac, Richard C Stevens, Shreedharan Sriram, Ronghua Chen, Carol A Rohl, Jason M Johnson, and Tomas Babak. “A quantitative atlas of polyadenylation in five mammals.” en. In: *Genome Res.* 22.6 (June 2012), pp. 1173–1183.

[16] Georges Martin, Andreas R Gruber, Walter Keller, and Mihaela Zavolan. “Genome-wide analysis of pre-mRNA 3′ end processing reveals a decisive role of human cleavage factor I in the regulation of 3′ UTR length.” en. In: *Cell Rep.* 1.6 (June 2012), pp. 753–763.

[17] Mathias Jenal et al. “The poly(A)-binding protein nuclear 1 suppresses alternative cleavage and polyadenylation sites.” en. In: *Cell* 149.3 (Apr. 2012), pp. 538–553.

[18] Nicholas T Ingolia, Sina Ghaemmaghami, John R S Newman, and Jonathan S Weissman. “Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling.” en. In: *Science* 324.5924 (Apr. 2009), pp. 218–223.

[19] Zachary D Stephens, Skylar Y Lee, Faraz Faghri, Roy H Campbell, Chengxiang Zhai, Miles J Efron, Ravishankar Iyer, Michael C Schatz, Saurabh Sinha, and Gene E Robinson. “Big Data: Astronomical or Genomical?” en. In: *PLoS Biol.* 13.7 (July 2015), e1002195.

[20] Rasko Leinonen, Hideaki Sugawara, Martin Shumway, and International Nucleotide Sequence Database Collaboration. “The sequence read archive.” en. In: *Nucleic Acids Res.* 39.Database issue (Jan. 2011), pp. D19–21.

[21] Tanya Barrett et al. “NCBI GEO: archive for functional genomics data sets–update.” en. In: *Nucleic Acids Res.* 41.Database issue (Jan. 2013), pp. D991–5.

[22] Ana Kozomara and Sam Griffiths-Jones. “miRBase: integrating microRNA annotation and deep-sequencing data.” In: *Nucleic Acids Res.* 39.suppl 1 (Jan. 2011), pp. D152–D157.

[23] Hadi Jorjani, Stephanie Kehr, Dominik J Jedlinski, Rafal Gumienny, Jana Hertel, Peter F Stadler, Mihaela Zavolan, and Andreas R Gruber. “An updated human snoRNAome.” en. In: *Nucleic Acids Res.* 44.11 (June 2016), pp. 5068–5082.

[24] The RNAcentral Consortium. “RNAcentral: a comprehensive database of non-coding RNA sequences.” en. In: *Nucleic Acids Res.* 45.D1 (Jan. 2017), pp. D128–D134.

[25] Paul Julian Kersey et al. “Ensembl Genomes 2016: more genomes, more complexity.” en. In: *Nucleic Acids Res.* 44.D1 (Jan. 2016), pp. D574–80.

[26] Jennifer Harrow et al. “GENCODE: the reference human genome annotation for The ENCODE Project.” en. In: *Genome Res.* 22.9 (Sept. 2012), pp. 1760–1774.

[27] B Gruning, A Guerler, J Hillman-Jackson, and others. “The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update.” In: *Nucleic acids* (2016).

[28] Mohsen Khorshid, Christoph Rodak, and Mihaela Zavolan. “CLIPZ: a database and analysis environment for experimentally determined binding sites of RNA-binding proteins.” en. In: *Nucleic Acids Res.* 39.Database issue (Jan. 2011), pp. D245–52.

[29] Kristian Ovaska et al. “Large-scale data integration framework provides a comprehensive view on glioblastoma multiforme.” In: *Genome Med.* 2.9 (Sept. 2010), p. 65.

[30] J Köster and S Rahmann. “Snakemake—a scalable bioinformatics workflow engine.” In: *Bioinformatics* (2012).

[31] Peter Amstutz et al. *Common Workflow Language, v1.0*. July 2016.

[32] Shareen A Iqbal, Joshua D Wallach, Muin J Khoury, Sheri D Schully, and John P A Ioannidis. “Reproducible Research Practices and Transparency across the Biomedical Literature.” en. In: *PLoS Biol.* 14.1 (Jan. 2016), e1002333.

[33] Ana Conesa et al. “A survey of best practices for RNA-seq data analysis.” en. In: *Genome Biol.* 17 (Jan. 2016), p. 13.

[34] Peter J A Cock, Christopher J Fields, Naohisa Goto, Michael L Heuer, and Peter M Rice. “The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants.” In: *Nucleic Acids Res.* 38.6 (Dec. 2009), pp. 1767–1771.

[35] Simon Andrews and Others. *FastQC: a quality control tool for high throughput sequence data*. 2010.

[36] Timo Lassmann. “TagDust2: a generic method to extract reads from sequencing data.” en. In: *BMC Bioinformatics* 16 (Jan. 2015), p. 24.

[37] L R Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition.” In: *Proc. IEEE* 77.2 (Feb. 1989), pp. 257–286.

[38] Marcel Martin. “Cutadapt removes adapter sequences from high-throughput sequencing reads.” In: *EMBnet.journal* 17.1 (May 2011), pp. 10–12.

[39] A Gordon and G J Hannon. “Fastx-toolkit.” In: *FASTQ/A short-reads preprocessing tools (unpublished) http://hannonlab. cshl. edu/-fastx_toolkit* 5 (2010).

[40] Tanja Magoč and Steven L Salzberg. "FLASH: fast length adjustment of short reads to improve genome assemblies." en. In: *Bioinformatics* 27.21 (Nov. 2011), pp. 2957–2963.

[41] Heng Li and Nils Homer. "A survey of sequence alignment algorithms for next-generation sequencing." In: *Brief. Bioinform.* 11.5 (May 2010), pp. 473–483.

[42] Dinesh P Mehta and Sartaj Sahni. *Handbook of Data Structures and Applications*. en. CRC Press, Oct. 2004.

[43] P Weiner. "Linear pattern matching algorithms." In: *14th Annual Symposium on Switching and Automata Theory (swat 1973)*. Oct. 1973, pp. 1–11.

[44] U Manber and G Myers. "Suffix Arrays: A New Method for On-Line String Searches." In: *SIAM J. Comput.* 22.5 (Oct. 1993), pp. 935–948.

[45] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. "The Sequence Alignment/Map format and SAMtools." In: *Bioinformatics* 25.16 (Aug. 2009), pp. 2078–2079.

[46] Markus Hsi-Yang Fritz, Rasko Leinonen, Guy Cochrane, and Ewan Birney. "Efficient storage of high throughput DNA sequencing data using reference-based compression." en. In: *Genome Res.* 21.5 (May 2011), pp. 734–740.

[47] Derek W Barnett, Erik K Garrison, Aaron R Quinlan, Michael P Strömberg, and Gabor T Marth. "BamTools: a C++ API and toolkit for analyzing and managing BAM files." en. In: *Bioinformatics* 27.12 (June 2011), pp. 1691–1692.

[48] James K Bonfield. "The Scramble conversion tool." en. In: *Bioinformatics* 30.19 (Oct. 2014), pp. 2818–2819.

[49] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." In: *Genome Biol.* 10.3 (Mar. 2009), R25.

[50] Cole Trapnell, Lior Pachter, and Steven L Salzberg. "TopHat: discovering splice junctions with RNA-Seq." In: *Bioinformatics* 25.9 (Mar. 2009), pp. 1105–1111.

[51] Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. "STAR: ultrafast universal RNA-seq aligner." In: *Bioinformatics* 29.1 (Jan. 2013), pp. 15–21.

[52] Steve Hoffmann, Christian Otto, Stefan Kurtz, Cynthia M Sharma, Philipp Khaitovich, Jörg Vogel, Peter F Stadler, and Jörg Hackermüller. "Fast mapping of short sequences with mismatches, insertions and deletions using index structures." en. In: *PLoS Comput. Biol.* 5.9 (Sept. 2009), e1000502.

[53] Steve Hoffmann et al. “A multi-split mapping algorithm for circular RNA, splicing, trans-splicing and fusion detection.” en. In: *Genome Biol.* 15.2 (Feb. 2014), R34.

[54] Nicolas L Bray, Harold Pimentel, Páll Melsted, and Lior Pachter. “Near-optimal probabilistic RNA-seq quantification.” en. In: *Nat. Biotechnol.* 34.5 (May 2016), pp. 525–527.

[55] Simon Anders, Paul Theodor Pyl, and Wolfgang Huber. “HTSeq–a Python framework to work with high-throughput sequencing data.” en. In: *Bioinformatics* 31.2 (Jan. 2015), pp. 166–169.

[56] Yang Liao, Gordon K Smyth, and Wei Shi. “featureCounts: an efficient general purpose program for assigning sequence reads to genomic features.” en. In: *Bioinformatics* 30.7 (Apr. 2014), pp. 923–930.

[57] Aaron R Quinlan and Ira M Hall. “BEDTools: a flexible suite of utilities for comparing genomic features.” In: *Bioinformatics* 26.6 (Mar. 2010), pp. 841–842.

[58] Dimos Gaidatzis, Anita Lerch, Florian Hahne, and Michael B Stadler. “QuasR: quantification and annotation of short reads in R.” en. In: *Bioinformatics* 31.7 (Apr. 2015), pp. 1130–1132.

[59] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. “edgeR: a Bioconductor package for differential expression analysis of digital gene expression data.” In: *Bioinformatics* 26.1 (Jan. 2010), pp. 139–140.

[60] Simon Anders and Wolfgang Huber. “Differential expression analysis for sequence count data.” In: *Genome Biol.* 11.10 (Oct. 2010), R106.

[61] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. “Mapping and quantifying mammalian transcriptomes by RNA-Seq.” en. In: *Nat. Methods* 5.7 (July 2008), pp. 621–628.

[62] Günter P Wagner, Koryu Kin, and Vincent J Lynch. “Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples.” In: *Theory Biosci.* 131.4 (Dec. 2012), pp. 281–285.

[63] A P Dempster, N M Laird, and D B Rubin. “Maximum Likelihood from Incomplete Data via the EM Algorithm.” In: *J. R. Stat. Soc. Series B Stat. Methodol.* 39.1 (1977), pp. 1–38.

[64] Lior Pachter. “Models for transcript quantification from RNA-Seq.” In: (Apr. 2011). arXiv: 1104.3889 [q-bio.GN].

[65] Y Xing, T Yu, Y N Wu, M Roy, J Kim, and others. “An expectation-maximization algorithm for probabilistic reconstructions of full-length isoforms from splice graphs.” In: *Nucleic acids* (2006).

[66] Tamara Steijger et al. “Assessment of transcript reconstruction methods for RNA-seq.” In: *Nat. Methods* 10.12 (Nov. 2013), pp. 1177–1184.

[67] Katharina E Hayer, Pizarro Angel, Nicholas F Lahens, John B Hogenesch, and Gregory R Grant. "Benchmark Analysis of Algorithms for Determining and Quantifying Full-length mRNA Splice Forms from RNA-Seq Data." In: *Bioinformatics* (2015), btv488.

[68] Steffen Heber, Max Alekseyev, Sing-Hoi Sze, Haixu Tang, and Pavel A Pevzner. "Splicing graphs and EST assembly problem." en. In: *Bioinformatics* 18 Suppl 1 (2002), S181–8.

[69] Susan M Berget, Claire Moore, and Phillip A Sharp. "Spliced segments at the 5' terminus of adenovirus 2 late mRNA." In: *Proceedings of the National Academy of Sciences* 74.8 (Aug. 1977), pp. 3171–3175.

[70] L T Chow, R E Gelinas, T R Broker, and R J Roberts. "An amazing sequence arrangement at the 5' ends of adenovirus 2 messenger RNA." en. In: *Cell* 12.1 (Sept. 1977), pp. 1–8.

[71] P A Sharp. "Split genes and RNA splicing." en. In: *Cell* 77.6 (June 1994), pp. 805–815.

[72] Hagen Tilgner, David G Knowles, Rory Johnson, Carrie A Davis, Sudipto Chakrabortty, Sarah Djebali, Joao Curado, Michael Snyder, Thomas R Gingeras, and Roderic Guigó. "Deep sequencing of subcellular RNA fractions shows splicing to be predominantly co-transcriptional in the human genome but inefficient for lncRNAs." en. In: *Genome Res.* 22.9 (Sept. 2012), pp. 1616–1625.

[73] Jay R Hesselberth. "Lives that introns lead after splicing." In: *Wiley Interdiscip. Rev. RNA* 4.6 (July 2013), pp. 677–691.

[74] Qun Pan, Ofer Shai, Leo J Lee, Brendan J Frey, and Benjamin J Blencowe. "Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing." en. In: *Nat. Genet.* 40.12 (Dec. 2008), pp. 1413–1415.

[75] Gael P Alamancos, Eneritz Agirre, and Eduardo Eyras. "Methods to study splicing from high-throughput RNA sequencing data." In: *Methods Mol. Biol.* 1126 (2014), pp. 357–397.

[76] Eric T Wang, Rickard Sandberg, Shujun Luo, Irina Khrebtukova, Lu Zhang, Christine Mayr, Stephen F Kingsmore, Gary P Schroth, and Christopher B Burge. "Alternative isoform regulation in human tissue transcriptomes." In: *Nature* 456.7221 (Nov. 2008), pp. 470–476.

[77] Yarden Katz, Eric T Wang, Edoardo M Airoldi, and Christopher B Burge. "Analysis and design of RNA sequencing experiments for identifying isoform regulation." In: *Nat. Methods* 7.12 (Nov. 2010), pp. 1009–1015.

[78] Juw Won Park, Collin Tokheim, Shihao Shen, and Yi Xing. "Identifying differential alternative splicing events from RNA sequencing data using RNASeq-MATS." In: *Methods Mol. Biol.* 1038 (2013), pp. 171–179.

[79] Shihao Shen, Juw Won Park, Zhi-Xiang Lu, Lan Lin, Michael D Henry, Ying Nian Wu, Qing Zhou, and Yi Xing. "rMATS: Robust and flexible detection of differential alternative splicing from replicate RNA-Seq data." In: *Proc. Natl. Acad. Sci. U. S. A.* (Dec. 2014).

[80] Gael P Alamancos, Amadís Pagès, Juan L Trincado, Nicolás Bellora, and Eduardo Eyras. "Leveraging transcript quantification for fast computation of alternative splicing profiles." In: *RNA* 21.9 (Sept. 2015), pp. 1521–1531.

[81] Tzu-Ming Chern, Erik van Nimwegen, Chikatoshi Kai, Jun Kawai, Piero Carninci, Yoshihide Hayashizaki, and Mihaela Zavolan. "A simple physical model predicts small exon length variations." In: *PLoS Genet.* 2.4 (Apr. 2006), e45.

[82] Manuel Irimia et al. "A highly conserved program of neuronal microexons is misregulated in autistic brains." In: *Cell* 159.7 (Dec. 2014), pp. 1511–1523.

[83] Jorge Vaquero-Garcia, Alejandro Barrera, Matthew R Gazzara, Juan González-Vallinas, Nicholas F Lahens, John B Hogenesch, Kristen W Lynch, and Yoseph Barash. "A new view of transcriptome complexity and regulation through the lens of local splicing variations." en. In: *Elife* 5 (Feb. 2016), e11752.

[84] Javier Tapial et al. "An atlas of alternative splicing profiles and functional associations reveals new regulatory programs and genes that simultaneously express multiple major isoforms." en. In: *Genome Res.* (Aug. 2017).

[85] Ulrich Braunschweig, Nuno L Barbosa-Morais, Qun Pan, Emil N Nachman, Babak Alipanahi, Thomas Gonatopoulos-Pournatzis, Brendan Frey, Manuel Irimia, and Benjamin J Blencowe. "Widespread intron retention in mammals functionally tunes transcriptomes." In: *Genome Res.* (Sept. 2014).

[86] Robert Middleton et al. "IRFinder: assessing the impact of intron retention on mammalian gene expression." en. In: *Genome Biol.* 18.1 (Mar. 2017), p. 51.

[87] Harold Pimentel, John G Conboy, and Lior Pachter. "Keep Me Around: Intron Retention Detection and Analysis." In: (Oct. 2015). arXiv: 1510.00696 [q-bio.GN].

[88] Yang Bai, Shufan Ji, and Yadong Wang. "IRcall and IRclassifier: two methods for flexible detection of intron retention events from RNA-Seq data." In: *BMC Genomics* 16 Suppl 2 (Jan. 2015), S9.

[89] A J Shatkin and J L Manley. "The ends of the affair: capping and polyadenylation." en. In: *Nat. Struct. Biol.* 7.10 (Oct. 2000), pp. 838–842.

[90] Tamar Juven-Gershon and James T Kadonaga. "Regulation of gene expression via the core promoter and the basal transcriptional machinery." en. In: *Dev. Biol.* 339.2 (Mar. 2010), pp. 225–229.

[91] Shivendra Kishore, Sandra Luber, and Mihaela Zavolan. "Deciphering the role of RNA-binding proteins in the post-transcriptional control of gene expression." en. In: *Brief. Funct. Genomics* 9.5-6 (Dec. 2010), pp. 391–404.

[92] Nick J Proudfoot. "Ending the message: poly(A) signals then and now." en. In: *Genes Dev.* 25.17 (Sept. 2011), pp. 1770–1782.

[93] G Edwalds-Gilbert, K L Veraldi, and C Milcarek. "Alternative poly(A) site selection in complex transcription units: means to an end?" en. In: *Nucleic Acids Res.* 25.13 (July 1997), pp. 2547–2561.

[94] Andreas J Gruber, Ralf Schmidt, Andreas R Gruber, Georges Martin, Souvik Ghosh, Manuel Belmadani, Walter Keller, and Mihaela Zavolan. "A comprehensive analysis of 3′ end sequencing data sets reveals novel polyadenylation signals and the repressive role of heterogeneous ribonucleoprotein C on cleavage and polyadenylation." en. In: *Genome Res.* 26.8 (Aug. 2016), pp. 1145–1159.

[95] Georges Martin, Ralf Schmidt, Andreas J Gruber, Souvik Ghosh, Walter Keller, and Mihaela Zavolan. *3′ End Sequencing Library Preparation with A-seq2 Protocol*. https://www.jove.com/video/56129/3-end-sequencing-library-preparation-with-a-seq2. Accessed: 2017-9-26.

[96] J Michael Brockman, Priyam Singh, Donglin Liu, Sean Quinlan, Jesse Salisbury, and Joel H Graber. "PACdb: PolyA Cleavage Site and 3′-UTR Database." en. In: *Bioinformatics* 21.18 (Sept. 2005), pp. 3691–3693.

[97] Ju Youn Lee, Ijen Yeh, Ji Yeon Park, and Bin Tian. "PolyA_DB 2: mRNA polyadenylation sites in vertebrate genes." en. In: *Nucleic Acids Res.* 35.Database issue (Jan. 2007), pp. D165–8.

[98] Sören Müller et al. "APADB: a database for alternative polyadenylation and microRNA regulation events." en. In: *Database* 2014 (July 2014).

[99] Chioniso P Masamha, Zheng Xia, Jingxuan Yang, Todd R Albrecht, Min Li, Ann-Bin Shyu, Wei Li, and Eric J Wagner. "CFIm25 links alternative polyadenylation to glioblastoma tumour suppression." en. In: *Nature* 510.7505 (June 2014), pp. 412–416.

[100] Zheng Xia, Lawrence A Donehower, Thomas A Cooper, Joel R Neilson, David A Wheeler, Eric J Wagner, and Wei Li. "Dynamic analyses of alternative polyadenylation from RNA-seq reveal a 3′-UTR landscape across seven tumour types." en. In: *Nat. Commun.* 5 (Nov. 2014), p. 5274.

[101] Andreas J Gruber, Ralf Schmidt, Souvik Ghosh, Georges Martin, Andreas R Gruber, Erik van Nimwegen, and Mihaela Zavolan. "Discovery of global regulators of 3′ untranslated region processing in cancers with KAPAC." en. Sept. 2017.

[102] Allison C Mallory and Alena Shkumatava. “LncRNAs in vertebrates: advances and challenges.” en. In: *Biochimie* 117 (Oct. 2015), pp. 3–14.

[103] Li Yang, Michael O Duff, Brenton R Graveley, Gordon G Carmichael, and Ling-Ling Chen. “Genomewide characterization of non-polyadenylated RNAs.” In: *Genome Biol.* 12.2 (Feb. 2011), R16.

[104] Igor Ulitsky and David P Bartel. “lincRNAs: genomics, evolution, and mechanisms.” en. In: *Cell* 154.1 (July 2013), pp. 26–46.

[105] Igor Ulitsky, Alena Shkumatava, Calvin H Jan, Hazel Sive, and David P Bartel. “Conserved function of lincRNAs in vertebrate embryonic development despite rapid sequence evolution.” en. In: *Cell* 147.7 (Dec. 2011), pp. 1537–1550.

[106] Mitchell Guttman et al. “Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs.” en. In: *Nat. Biotechnol.* 28.5 (May 2010), pp. 503–510.

[107] Thomas Derrien et al. “The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression.” en. In: *Genome Res.* 22.9 (Sept. 2012), pp. 1775–1789.

[108] Paulo P Amaral, Michael B Clark, Dennis K Gascoigne, Marcel E Dinger, and John S Mattick. “lncRNAdb: a reference database for long noncoding RNAs.” en. In: *Nucleic Acids Res.* 39.Database issue (Jan. 2011), pp. D146–51.

[109] Dechao Bu et al. “NONCODE v3.0: integrative annotation of long noncoding RNAs.” en. In: *Nucleic Acids Res.* 40.Database issue (Jan. 2012), pp. D210–5.

[110] Pieter-Jan Volders, Kenny Helsens, Xiaowei Wang, Björn Menten, Lennart Martens, Kris Gevaert, Jo Vandesompele, and Pieter Mestdagh. “LNCipedia: a database for annotated human lncRNA transcript sequences and structures.” en. In: *Nucleic Acids Res.* 41.Database issue (Jan. 2013), pp. D246–51.

[111] Siyu Han, Yanchun Liang, Ying Li, and Wei Du. “Long Noncoding RNA Identification: Comparing Machine Learning Based Tools for Long Noncoding Transcripts Discrimination.” en. In: *Biomed Res. Int.* 2016 (Nov. 2016), p. 8496165.

[112] W James Kent, Charles W Sugnet, Terrence S Furey, Krishna M Roskin, Tom H Pringle, Alan M Zahler, and David Haussler. “The human genome browser at UCSC.” en. In: *Genome Res.* 12.6 (June 2002), pp. 996–1006.

[113] James T Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S Lander, Gad Getz, and Jill P Mesirov. “Integrative genomics viewer.” en. In: *Nat. Biotechnol.* 29.1 (Jan. 2011), pp. 24–26.

[114] Florian Hahne and Robert Ivanek. "Visualizing Genomic Data Using Gviz and Bioconductor." en. In: *Methods Mol. Biol.* 1418 (2016), pp. 335–351.

[115] Yarden Katz, Eric T Wang, Jacob Stilterra, Schraga Schwartz, Bang Wong, Helga Thorvaldsdóttir, James T Robinson, Jill P Mesirov, Edoardo M Airoldi, and Christopher B Burge. "Sashimi plots: Quantitative visualization of alternative isoform expression from RNA-seq data." Feb. 2014.

[116] Barmak Modrek and Christopher Lee. "A genomic view of alternative splicing." In: *Nat. Genet.* 30.1 (Jan. 2002), pp. 13–19.

[117] Mihaela Zavolan, Shinji Kondo, Christian Schonbach, Jun Adachi, David A Hume, Yoshihide Hayashizaki, Terry Gaasterland, RIKEN GER Group, and GSL Members. "Impact of alternative initiation, splicing, and termination on the diversity of the mRNA transcripts encoded by the mouse transcriptome." In: *Genome Res.* 13.6B (June 2003), pp. 1290–1300.

[118] Hideki Nagasaki, Masanori Arita, Tatsuya Nishizawa, Makiko Suwa, and Osamu Gotoh. "Species-specific variation of alternative splicing and transcriptional initiation in six eukaryotes." In: *Gene* 364 (2005), pp. 53–62.

[119] Robert K Bradley, Jason Merkin, Nicole J Lambert, and Christopher B Burge. "Alternative splicing of RNA triplets is often regulated and accelerates proteome evolution." In: *PLoS Biol.* 10.1 (Jan. 2012), e1001229.

[120] Lauren M Reinke, Yilin Xu, and Chonghui Cheng. "Snail represses the splicing regulator epithelial splicing regulatory protein 1 to promote epithelial-mesenchymal transition." In: *J. Biol. Chem.* 287.43 (2012), pp. 36435–36442.

[121] Mo Chen, Jian Zhang, and James L Manley. "Turning on a fuel switch of cancer: hnRNP proteins regulate alternative splicing of pyruvate kinase mRNA." In: *Cancer Res.* 70.22 (2010), pp. 8977–8980.

[122] Yoseph Barash, John A Calarco, Weijun Gao, Qun Pan, Xinchen Wang, Ofer Shai, Benjamin J Blencowe, and Brendan J Frey. "Deciphering the splicing code." In: *Nature* 465.7294 (2010), pp. 53–59.

[123] Mohini Jangi and Phillip A Sharp. "Building Robust Transcriptomes with Master Splicing Factors." In: *Cell* 159.3 (2014), pp. 487–498.

[124] Hong Han et al. "MBNL proteins repress ES-cell-specific alternative splicing and reprogramming." In: *Nature* 498.7453 (2013), pp. 241–245.

[125] Claude C Warzecha, Peng Jiang, Karine Amirikian, Kimberly A Dittmar, Hezhe Lu, Shihao Shen, Wei Guo, Yi Xing, and Russ P Carstens. "An ESRP-regulated splicing programme is abrogated during the epithelial-mesenchymal transition." In: *EMBO J.* 29.19 (2010), pp. 3286–3300.

[126] Michael L Tress et al. “The implications of alternative splicing in the ENCODE protein complement.” In: *Proc. Natl. Acad. Sci. U. S. A.* 104.13 (2007), pp. 5495–5500.

[127] Nuno L Barbosa-Morais et al. “The evolutionary landscape of alternative splicing in vertebrate species.” In: *Science* 338.6114 (2012), pp. 1587–1593.

[128] Ron Milo, Paul Jorgensen, Uri Moran, Griffin Weber, and Michael Springer. “BioNumbers—the database of key numbers in molecular and cell biology.” In: *Nucleic acids research* 38.suppl_1 (2009), pp. D750–D753.

[129] Kin Fai Au et al. “Characterization of the human ESC transcriptome by hybrid sequencing.” In: *Proc. Natl. Acad. Sci. U. S. A.* 110.50 (2013), E4821–30.

[130] Dominic Grün, Lennart Kester, and Alexander van Oudenaarden. “Validation of noise models for single-cell transcriptomics.” In: *Nat. Methods* 11.6 (June 2014), pp. 637–640.

[131] Pär G Engström et al. “Systematic evaluation of spliced alignment programs for RNA-seq data.” In: *Nat. Methods* November (Nov. 2013), pp. 10–12.

[132] Raghu Chandramohan, Po-Yen Wu, John H Phan, and May D Wang. “Benchmarking RNA-Seq quantification tools.” In: *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 2013 (2013), pp. 647–650.

[133] MAQC Consortium et al. “The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements.” In: *Nat. Biotechnol.* 24.9 (Sept. 2006), pp. 1151–1161.

[134] Igor Ulitsky, Alena Shkumatava, Calvin H Jan, Alexander O Subtelny, David Koppstein, George W Bell, Hazel Sive, and David P Bartel. “Extensive alternative polyadenylation during zebrafish development.” In: *Genome Res.* 22.10 (Oct. 2012), pp. 2054–2066.

[135] Peter Glaus, Antti Honkela, and Magnus Rattray. “Identifying differentially expressed transcripts from RNA-seq data with biological variation.” In: *Bioinformatics* 28.13 (2012), pp. 1721–1728.

[136] Panagiotis Papastamoulis, James Hensman, Peter Glaus, and Magnus Rattray. “Improved variational Bayes inference for transcript expression estimation.” In: *Stat. Appl. Genet. Mol. Biol.* 13.2 (2014), pp. 203–216.

[137] Wei Li and Tao Jiang. “Transcriptome assembly and isoform expression level estimation from biased RNA-Seq reads.” In: *Bioinformatics* 28.22 (2012), pp. 2914–2921.

[138] Cole Trapnell, Brian A Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J van Baren, Steven L Salzberg, Barbara J Wold, and Lior Pachter. "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation." In: *Nat. Biotechnol.* 28.5 (2010), pp. 511–515.

[139] Adam Roberts and Lior Pachter. "Streaming fragment assignment for real-time analysis of sequencing experiments." In: *Nat. Methods* 10.1 (2012), pp. 71–73.

[140] Marius Nicolae, Serghei Mangul, Ion I Măndoiu, and Alex Zelikovsky. "Estimation of alternative splicing isoform frequencies from RNA-Seq data." In: *Algorithms Mol. Biol.* 6.1 (2011), p. 9.

[141] Ernest Turro, Shu-Yi Su, Ângela Gonçalves, Lachlan J M Coin, Sylvia Richardson, and Alex Lewin. "Haplotype and isoform specific expression estimation using multi-mapping RNA-seq reads." In: *Genome Biol.* 12.2 (2011), R13.

[142] Bo Li, Victor Ruotti, Ron M Stewart, James A Thomson, and Colin N Dewey. "RNA-Seq gene expression estimation with read mapping uncertainty." In: *Bioinformatics* 26.4 (2009), pp. 493–500.

[143] Bo Li and Colin N Dewey. "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome." In: *BMC Bioinformatics* 12 (2011), p. 323.

[144] Hui Jiang and Wing Hung Wong. "Statistical inferences for isoform expression in RNA-Seq." In: *Bioinformatics* 25.8 (2009), pp. 1026–1032.

[145] Rob Patro, Stephen M Mount, and Carl Kingsford. "Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms." In: *Nat. Biotechnol.* 32.5 (May 2014), pp. 462–464.

[146] Naoki Nariai, Osamu Hirose, Kaname Kojima, and Masao Nagasaki. "TIGAR: transcript isoform abundance estimation method with gapped alignment of RNA-Seq data by variational Bayesian inference." In: *Bioinformatics* 29.18 (2013), pp. 2292–2299.

[147] N Nariai, K Kojima, T Mimori, Y Sato, Y Kawai, and others. "TIGAR2: sensitive and accurate estimation of transcript isoform expression with longer RNA-Seq reads." In: *BMC Genomics* 15.Suppl 10 (2014), S5.

[148] Thasso Griebel, Benedikt Zacher, Paolo Ribeca, Emanuele Raineri, Vincent Lacroix, Roderic Guigó, and Michael Sammeth. "Modelling and simulating generic RNA-Seq experiments with the flux simulator." In: *Nucleic Acids Res.* 40.20 (2012), pp. 10073–10083.

[149] Sophie Schbath, Véronique Martin, Matthias Zytnicki, Julien Fayolle, Valentin Loux, and Jean-François Gibrat. "Mapping reads on a genomic sequence: an algorithmic overview and a practical comparative analysis." In: *J. Comput. Biol.* 19.6 (June 2012), pp. 796–813.

[150] Ayat Hatem, Doruk Bozdağ, Amanda E Toland, and Ümit V Çatalyürek. "Benchmarking short sequence mapping tools." In: *BMC Bioinformatics* 14 (2013), p. 184.

[151] Andrew H Beck, Ziming Weng, Daniela M Witten, Shirley Zhu, Joseph W Foley, Phil Lacroute, Cheryl L Smith, Robert Tibshirani, Matt Van De Rijn, Arend Sidow, et al. "3'-end sequencing for expression quantification (3SEQ) from archival tumor samples." In: *PloS one* 5.1 (2010), e8768.

[152] Stefan Wilkening, Vicent Pelechano, Aino I Järvelin, Manu M Tekkedil, Simon Anders, Vladimir Benes, and Lars M Steinmetz. "An efficient method for genome-wide polyadenylation site mapping and RNA quantification." In: *Nucleic acids research* 41.5 (2013), e65–e65.

[153] Mainul Hoque, Wencheng Li, and Bin Tian. "Accurate mapping of cleavage and polyadenylation sites by 3' region extraction and deep sequencing." In: *Polyadenylation: Methods and Protocols* (2014), pp. 119–129.

[154] Claudia Angelini, Daniela De Canditiis, and Italia De Feis. "Computational approaches for isoform detection and estimation: good and bad news." In: *BMC Bioinformatics* 15 (2014), p. 135.

[155] Thomas J Hardcastle and Krystyna A Kelly. "baySeq: empirical Bayesian methods for identifying differential expression in sequence count data." In: *BMC Bioinformatics* 11 (2010), p. 422.

[156] Likun Wang, Zhixing Feng, Xi Wang, Xiaowo Wang, and Xuegong Zhang. "DEGseq: an R package for identifying differentially expressed genes from RNA-seq data." In: *Bioinformatics* 26.1 (2009), pp. 136–138.

[157] Franck Rapaport, Raya Khanin, Yupu Liang, Mono Pirun, Azra Krek, Paul Zumbo, Christopher E Mason, Nicholas D Socci, and Doron Betel. "Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data." In: *Genome Biol.* 14.9 (2013), R95.

[158] Cole Trapnell, David G Hendrickson, Martin Sauvageau, Loyal Goff, John L Rinn, and Lior Pachter. "Differential analysis of gene regulation at transcript resolution with RNA-seq." In: *Nat. Biotechnol.* 31.1 (Jan. 2013), pp. 46–53.

[159] Sahar Al Seesi, Yvette Tiagueu, Alexander Zelikovsky, and Ion Mandoiu. *Bootstrap-based differential gene expression analysis for RNA-Seq data with and without replicates*. 2014.

[160] Ernest Turro, William J Astle, and Simon Tavaré. "Flexible analysis of RNA-seq data using mixed effects models." In: *Bioinformatics* 30.2 (2014), pp. 180–188.

[161] N Leng, J Dawson, and C Kendziorski. "EBSeq: An R package for differential expression analysis using RNA-seq data." In: (2013).

[162] Toshiyuki Shiraki et al. "Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage." In: *Proc. Natl. Acad. Sci. U. S. A.* 100.26 (2003), pp. 15776–15781.

[163] SEQC/MAQC-III Consortium. "A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium." In: *Nat. Biotechnol.* 32.9 (Sept. 2014), pp. 903–914.

[164] Fiona Cunningham et al. "Ensembl 2015." In: *Nucleic Acids Res.* (2014).

[165] U Schneider, H U Schwenk, and G Bornkamm. "Characterization of EBV-genome negative "null" and "T" cell lines derived from children with acute lymphoblastic leukemia and leukemic transformed non-Hodgkin lymphoma." In: *Int. J. Cancer* 19.5 (1977), pp. 621–626.

[166] G J Todaro and H Green. "Quantitative studies of the growth of mouse embryo cells in culture and their development into established lines." In: *J. Cell Biol.* 17 (May 1963), pp. 299–313.

[167] Andreas R Gruber et al. "Global 3′ UTR shortening has a limited effect on protein abundance in proliferating T cells." In: *Nat. Commun.* 5 (2014), p. 5465.

[168] Daehwan Kim, Geo Pertea, Cole Trapnell, Harold Pimentel, Ryan Kelley, and Steven L Salzberg. "TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions." In: *Genome Biol.* 14.4 (2013), R36.

[169] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." In: *Genome Biol.* 10.3 (Jan. 2009), R25.

[170] Michael Lawrence, Robert Gentleman, and Vincent Carey. "rtracklayer: an R package for interfacing with genome browsers." In: *Bioinformatics* 25.14 (2009), pp. 1841–1842.

[171] Michael Lawrence, Wolfgang Huber, Hervé Pagès, Patrick Aboyoun, Marc Carlson, Robert Gentleman, Martin T Morgan, and Vincent J Carey. "Software for computing and annotating genomic ranges." In: *PLoS Comput. Biol.* 9.8 (2013), e1003118.

[172] Jean Hausser and Mihaela Zavolan. "Identification and consequences of miRNA–target interactions—beyond repression of gene expression." In: *Nature Reviews Genetics* 15.9 (2014), p. 599.

[173] Rickard Sandberg, Joel R Neilson, Arup Sarma, Phillip A Sharp, and Christopher B Burge. "Proliferating Cells Express mRNAs with Shortened 3' Untranslated Regions and Fewer MicroRNA Target Sites." en. In: *Science* 320.5883 (2008), pp. 1643–1647.

[174] Brad Lackford et al. "Fip1 regulates mRNA alternative polyadenylation to promote stem cell self-renewal." en. In: *EMBO J.* 33.8 (2014), pp. 878–889.

[175] Christine Mayr and David P. Bartel. "Widespread Shortening of 3'UTRs by Alternative Cleavage and Polyadenylation Activates Oncogenes in Cancer Cells." In: *Cell* 138.4 (Aug. 2009), pp. 673–684.

[176] Noah Spies, Christopher B Burge, and David P Bartel. "3' UTR-isoform choice has limited influence on the stability and translational efficiency of most mRNAs in mouse fibroblasts." In: *Genome Res.* 23.12 (Dec. 2013), pp. 2078–2090.

[177] Mireya Plass, Simon H Rasmussen, and Anders Krogh. "Highly accessible AU-rich regions in 3' untranslated regions are hotspots for binding of regulatory factors." en. In: *PLoS Comput. Biol.* 13.4 (Apr. 2017), e1005460.

[178] Yuefeng Lin et al. "An in-depth map of polyadenylation sites in cancer." In: *Nucleic Acids Res.* 40.17 (2012), pp. 8460–8471.

[179] Bin Tian, Jun Hu, Haibo Zhang, and Carol S Lutz. "A large-scale analysis of mRNA polyadenylation of human and mouse genes." In: *Nucleic Acids Res.* 33.1 (2005), pp. 201–212.

[180] Steve Lianoglou, Vidur Garg, Julie L Yang, Christina S Leslie, and Christine Mayr. "Ubiquitously transcribed genes use alternative polyadenylation to achieve tissue-specific expression." In: *Genes Dev.* 27.21 (2013), pp. 2380–2396.

[181] Nian Liu, Qing Dai, Guanqun Zheng, Chuan He, Marc Parisien, and Tao Pan. "N(6)-methyladenosine-dependent RNA structural switches regulate RNA-protein interactions." In: *Nature* 518.7540 (2015), pp. 560–564.

[182] Lorenzo Calviello, Neelanjan Mukherjee, Emanuel Wyler, Henrik Zauber, Antje Hirsekorn, Matthias Selbach, Markus Landthaler, Benedikt Obermayer, and Uwe Ohler. "Detecting actively translated open reading frames in ribosome profiling data." en. In: *Nat. Methods* 13.2 (Feb. 2016), pp. 165–170.

[183] Mihaela Pertea, Geo M Pertea, Corina M Antonescu, Tsung-Cheng Chang, Joshua T Mendell, and Steven L Salzberg. "StringTie enables improved reconstruction of a transcriptome from RNA-seq reads." In: *Nat. Biotechnol.* 33.3 (Mar. 2015), pp. 290–295.

[184] J Lagarde, B Uszczynska-Ratajczak, S Carbonell, and others. "High-throughput annotation of full-length long noncoding RNAs with Capture Long-Read Sequencing (CLS)." In: *bioRxiv* (2017).

[185] Mathias Uhlén et al. "Proteomics. Tissue-based map of the human proteome." en. In: *Science* 347.6220 (2015), p. 1260419.

[186] S Alice Long, Jerill Thorpe, Hannah A DeBerg, Vivian Gersuk, James Eddy, Kristina M Harris, Mario Ehlers, Kevan C Herold, Gerald T Nepom, and Peter S Linsley. "Partial exhaustion of CD8 T cells and clinical response to teplizumab in new-onset type 1 diabetes." en. In: *Sci Immunol* 1.5 (Nov. 2016).

[187] Bronwen L Aken et al. "The Ensembl gene annotation system." en. In: *Database* 2016 (2016).

[188] Nicholas F Lahens et al. "IVT-seq reveals extreme bias in RNA sequencing." en. In: *Genome Biol.* 15.6 (June 2014), R86.

[189] Irene Gallego Romero, Athma A Pai, Jenny Tung, and Yoav Gilad. "RNA-seq: impact of RNA degradation on transcript quantification." In: *BMC biology* 12.1 (2014), p. 42.

[190] Nikolay Kolesnikov et al. "ArrayExpress update–simplifying data submissions." en. In: *Nucleic Acids Res.* 43.Database issue (Jan. 2015), pp. D1113–6.

[191] Soile Tuomela, Sini Rautio, Helena Ahlfors, Viveka Öling, Verna Salo, Ubaid Ullah, Zhi Chen, Saara Hämälistö, Subhash K Tripathi, Tarmo Äijö, et al. "Comparative analysis of human and mouse transcriptomes of Th17 cell priming." In: *Oncotarget* 7.12 (2016), p. 13416.

[192] A S Hinrichs et al. "The UCSC Genome Browser Database: update 2006." en. In: *Nucleic Acids Res.* 34.Database issue (2006), pp. D590–8.

[193] Leiming You et al. "APASdb: a database describing alternative poly(A) sites and selection of heterogeneous cleavage sites downstream of poly(A) signals." In: *Nucleic Acids Res.* 43.Database issue (Jan. 2015), pp. D59–67.

[194] Ryan K Dale, Brent S Pedersen, and Aaron R Quinlan. "Pybedtools: a flexible Python library for manipulating genomic datasets and annotations." en. In: *Bioinformatics* 27.24 (2011), pp. 3423–3424.

[195] S Van Der Walt, S C Colbert, and others. "The NumPy array: a structure for efficient numerical computation." In: *Comput. Sci. Eng.* (2011).

[196] Eric Jones, Travis Oliphant, and Pearu Peterson. "others. SciPy: Open source scientific tools for Python. 2001." In: *URL http://www.scipy. org* (2016).

[197] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *J. Mach. Learn. Res.* 12.Oct (2011), pp. 2825–2830.

[198] Wes McKinney. "pandas: a foundational Python library for data analysis and statistics." In: *Python for High Performance and Scientific Computing* 14 (2011).

[199] Rob Patro, Geet Duggal, Michael I Love, Rafael A Irizarry, and Carl Kingsford. "Salmon provides fast and bias-aware quantification of transcript expression." en. In: *Nat. Methods* (2017).

[200] Alexander Kanitz, Foivos Gypas, Andreas J Gruber, Andreas R Gruber, Georges Martin, and Mihaela Zavolan. "Comparative assessment of methods for the computational inference of transcript isoform abundance from RNA-seq data." en. In: *Genome Biol.* 16 (2015), p. 150.

[201] Mingxiang Teng et al. "A benchmark for RNA-seq quantification pipelines." en. In: *Genome Biol.* 17 (2016), p. 74.

[202] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. "Basic local alignment search tool." en. In: *J. Mol. Biol.* 215.3 (1990), pp. 403–410.

[203] Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. "BLAST+: architecture and applications." In: *BMC Bioinformatics* 10.1 (2009), p. 421.

[204] David R Bentley et al. "Accurate whole human genome sequencing using reversible terminator chemistry." en. In: *Nature* 456.7218 (Nov. 2008), pp. 53–59.

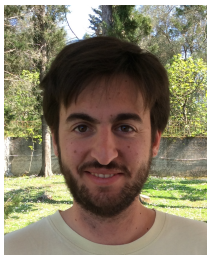
[205] Steffen Möller et al. "Robust Cross-Platform Workflows: How Technical and Scientific Communities Collaborate to Develop, Test and Share Best Practices for Data Analysis." en. In: *Data Sci. Eng.* (Nov. 2017), pp. 1–13.

[206] Charlotte Soneson and Mark D Robinson. "iCOBRA: open, reproducible, standardized and live method benchmarking." en. In: *Nat. Methods* 13.4 (Apr. 2016), p. 283.

[207] Andreas J Gruber, Foivos Gypas, Andrea Riba, Ralf Schmidt, and Mihaela Zavolan. "Terminal exon characterization with TECtool reveals an abundance of cell-specific isoforms." In: *Nature methods* 15.10 (2018), p. 832.

CV: FOIVOS GYPAS

E-mail: fgypas@gmail.com
Homepage: www.gypas.com/



POSITIONS

- Postdoctoral researcher (Bioinformatics) at Friedrich Miescher Institute for Biomedical Research since April 2019.
  Advisor: Helge Grosshans
- Postdoctoral researcher (Bioinformatics) at Biozentrum, University of Basel from April 2018 to March 2019.
  Advisor: Mihaela Zavolan
- PhD candidate at Biozentrum, University of Basel (Mihaela Zavolan group) from January 2014 to March 2018.
  Advisor: Mihaela Zavolan
- Teaching assistant for the course "Basic notions of computer programming" (python course), at Biozentrum, University of Basel for the winter semesters 2015, 2016, 2017 and 2018.
- Teaching assistant for the course "Bioinformatics" at University of Athens, Faculty of Biology, from March 2013 to August 2013.

EDUCATION

- PhD Bioinformatics, Biozentrum, University of Basel, Switzerland , 2018.
  Thesis: **Computational methods for the identification and quantification of transcript isoforms from next generation sequencing data**
  Group: RNA Regulatory Networks, Advisor: Mihaela Zavolan
- MSc Bioinformatics, department of Cell Biology and Biophysics, Faculty of Biology, University of Athens, Greece, 2013.
  Thesis: **Database of Molecular Recognition Features (MoRFs) in membrane proteins**
  Lab: Biophysics and Bioinformatics Laboratory, Supervisor: Stavros J. Hamodrakas
- Diploma (5-year course) in Electronic and Computer Engineering, department of Electronic and Computer Engineering at Technical University of Crete, Greece, 2011.
  Thesis: **Development of a Neural Model for Gene Analysis**
  Lab: Digital Image & Signal Processing Lab (DISPLAY), Supervisor: Michalis E. Zervakis
- Geitonas High School, 2005.

PUBLICATIONS

- Gruber AJ*, **Gypas F***, Riba A, Schmidt R, Zavolan M. **Terminal exon characterization with TECtool reveals an abundance of cell-specific isoforms.** Nat Methods. 2018 Sep 10. doi: 10.1038/s41592-018-0114-z. * equal contribution
- Gumienny R, Jedlinski DJ, Schmidt A, **Gypas F**, Martin G, Vina-Vilaseca A, Zavolan M. **High-throughput identification of C/D box snoRNA targets with CLIP and RiboMeth-seq.** Nucleic Acids Res. 2016 Dec 27. doi:10.1093/nar/gkw1321. PMID: 28031372.
- Mittal N, Kunz C, **Gypas F**, Kishore S, Martin G, Wenzel F, Nimwegen E, Schaer P, Zavolan M. **Ewing sarcoma breakpoint region 1 prevents transcription-associated genome instability.** bioRxiv (2015): 034215.
- Kanitz A*, **Gypas F***, Gruber AJ, Gruber AR, Martin G, Zavolan M. **Comparative assessment of methods for the computational inference of transcript isoform abundance from RNA-seq data.** Genome Biol. 2015 Jul 23;16(1):150. PMID:26201343. * equal contribution
- **Human microbiome and the methods of its study - metagenomics, F. Gypas**, A-F.A. Mentis. Acta Microbiologica Hellenica, April-June 2014, Volume 59, Issue 2.
- **Gypas F**, Tsaousis GN, Hamodrakas SJ. **mpMoRFsDB: a database of molecular recognition features in membrane proteins.** Bioinformatics. 2013 Oct 1;29(19):2517-8. doi: 10.1093/bioinformatics/btt427. Epub 2013 Jul 26. PubMed PMID: 23894139.
- **Human enteric microbiome: Its role in health and disease.** A.F.A. Mentis, **F. Gypas**, A.F. Mentis. Archives of Hellenic Medicine. May-June 2013, Volume 30, No 3.
- **Gypas F**, Bei ES, Zervakis M, Sfakianakis S. **A disease annotation study of gene signatures in a breast cancer microarray dataset.** Conf Proc IEEE Eng Med Biol Soc. 2011;2011:5551-4. doi: 10.1109/IEMBS.2011.6091416. PubMed PMID: 22255596

RESEARCH INTERESTS

- Bioinformatics and computational biology
- Next generation sequencing, transcriptomics, gene regulation, ncRNA, RNA-binding proteins, disordered proteins

TECHNICAL SKILLS

- Currently using programming languages: Python, R
- Used to program in: Java, C, Matlab, Perl, SQL
- Web Development: HTML, CSS, PHP, Javascript, Flask, Django, Angular
- Best practices: git, CI/CD, docker, kubernetes, singularity, conda, git
- Pipeline frameworks: Snakemake, CWL, Anduril
- Operating Systems: Mac, Linux, Windows
- Typesetting Tools: LaTeX, Office
- Graphic Arts Software: Adobe Photoshop, Inkscape

AWARDS / DISTINCTIONS

- J.C.W Shepherd Phd Student Prize for Scientific Excellence, 2018.
- Travel grand to attend and lead a project in the Biohackathon in Paris November 2018.
- Marie Curie PhD Fellow (project number 607720) from January 2014 to December 2017.
- NCCR RNA and Disease travel grant for the RNP and Disease meeting on 14-17 October 2015 in Marrakech, Morocco.
- Selected first for the graduate program in Bioinformatics for the academic year 2011-2012.
- Performance Fellowship by the Technical University of Crete for the academic year 2005-2006.
- Successful participation at mathematical and physics competitions organized by Hellenic Mathematic Society and Greek Physics Union from 2000-2005.

LANGUAGES

- Greek, Native
- English, Michigan Proficiency (ECPE)
- German (basic), Zertifikat (ZD)

ACTIVITIES AND INTERESTS

- Basketball, swimming.
- Traveling, cooking.

REFERENCES

- Available upon request.